

CURRICULUM
OF
COMPUTER SCIENCE, SOFTWARE ENGINEERING
AND INFORMATION TECHNOLOGY
BS
MS

(Revised 2009)



HIGHER EDUCATION COMMISSION
ISLAMABAD

CURRICULUM DIVISION, HEC

Dr. Syed Sohail H. Naqvi	Executive Director
Prof. Dr. Altaf Ali G. Shaikh	Member (Acad)
Miss Ghayyur Fatima	Director (Curri)
Mr. M. Tahir Ali Shah	Deputy Director (Curri)
Mr. Shafiullah Khan	Deputy Director

TABLE OF CONTENTS

1. Introduction.....	6
2. Computing	16
Computer Science	
Software Engineering	
Information Technology	
3. Structures for BS Programme.....	17
4. Course Contents (Computing-Core Courses).....	19
5. Computer Science Curricula – 2009.....	26
BS in Computer Science	31
MS in Computer Science.....	54
6. Software Engineering Curricula - 2009	72
BS in Software Engineering	
MS in Software Engineering	106
7. Information Technology Curricula - 2009	114
BS in Information Technology	
MS in Information Technology.....	124
8. Annexures – A, B, C, D & E.....	143

PREFACE

Curriculum of a subject is said to be the throbbing pulse of a nation. By looking at the curriculum one can judge the state of intellectual development and the state of progress of the nation. The world has turned into a global village; new ideas and information are pouring in like a stream. It is, therefore, imperative to update our curricula regularly by introducing the recent developments in the relevant fields of knowledge.

In exercise of the powers conferred by sub-section (1) of section 3 of the Federal Supervision of Curricula Textbooks and Maintenance of Standards of Education Act 1976, the Federal Government vide notification No. D773/76-JEA (cur.), dated December 4th 1976, appointed the University Grants Commission as the competent authority to look after the curriculum revision work beyond class XII at the bachelor level and onwards to all degrees, certificates and diplomas awarded by degree colleges, universities and other institutions of higher education.

In pursuance of the above decisions and directives, the Higher Education Commission (HEC) is continually performing curriculum revision in collaboration with universities. According to the decision of the special meeting of Vice-Chancellor's Committee, the curriculum of a subject must be reviewed after every 3 years.

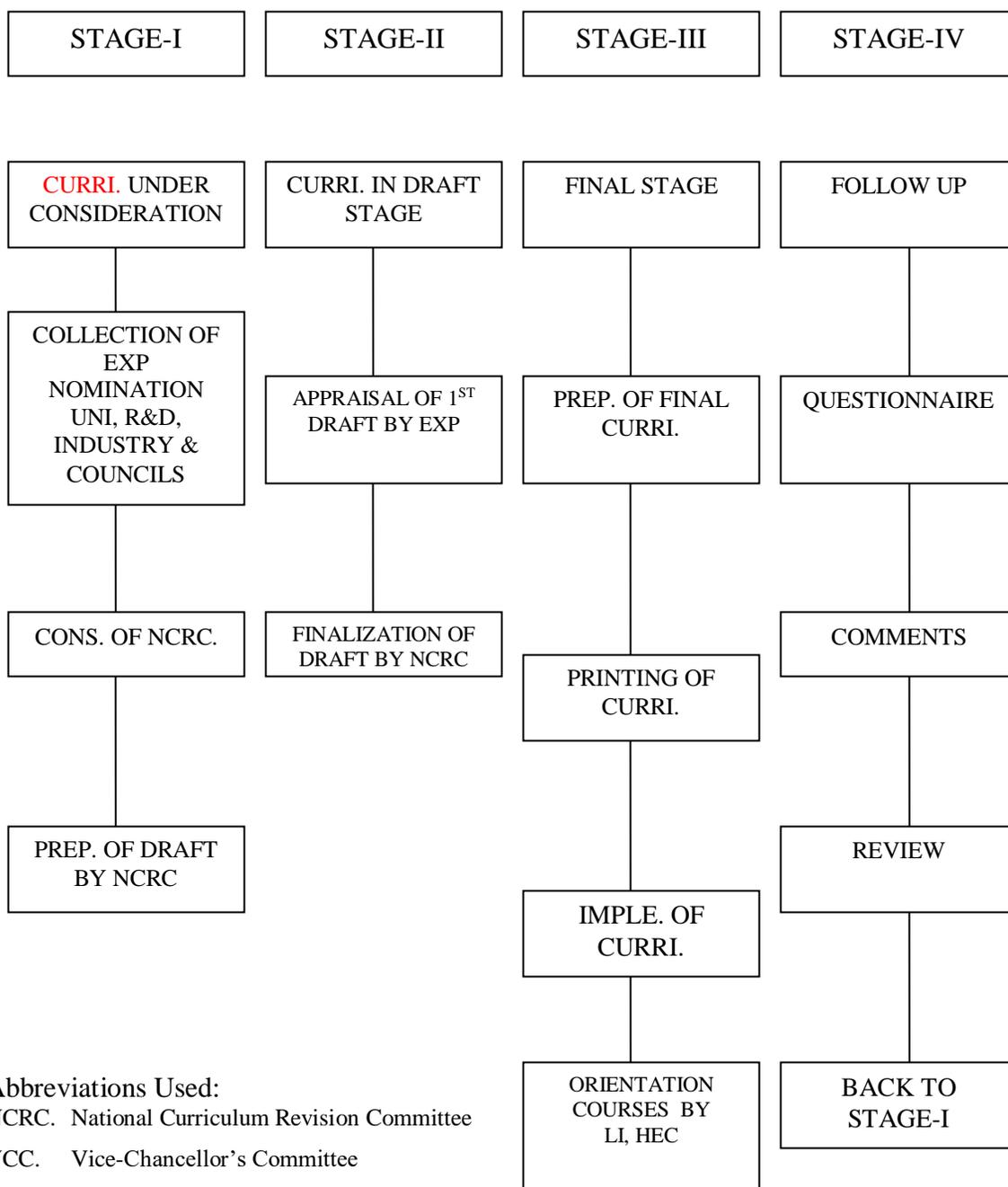
A committee of experts comprising of conveners from the National Curriculum Revision of HEC in Basic, Applied Social Sciences and Engineering disciplines met in April 2007 and developed a unified template to standardize degree programs in the country to bring the national curriculum at par with international standards, and to fulfill the needs of the local industries. It also aimed to give a basic, broad based knowledge to the students to ensure the quality of education. The new Bachelor (BS) degree shall be of 4 years duration, and will require the completion of 130-136 credit hours. The engineering degree will devote 65-70% of the curriculum towards engineering courses, and 35--30% to non Engineering courses.

For the purpose of curriculum revision various committees are constituted at the national level, comprising of senior teachers nominated by universities, degree awarding institutions, R&D organizations, respective accreditation councils and stake holders. The joint National Curriculum Revision Committee for Computer Science, Software Engineering and Information Technology in a meeting held on June 12-13, 2009 at the HEC Islamabad revised the curriculum in the light of the unified template. The revised draft curriculum is being circulated for implementation in the concerned institutions.

PROF. DR. ALTAF ALI G. SHAIKH
Member Academics

August 2009

CURRICULUM DEVELOPMENT



Abbreviations Used:

NCRC. National Curriculum Revision Committee

VCC. Vice-Chancellor's Committee

EXP. Experts

COL. Colleges

UNI. Universities

PREP. Preparation

REC. Recommendations

LI Learning Innovation

R&D Research & Development Organization

HEC Higher Education Commission

***National Joint Computing
(Computer Science, Software Engineering and
Information Technology)
Curriculum Committee
(NJCCC)***

Introduction

1. Computing Curricula Development-An Ongoing Activity

Computing is a dynamic and fast expanding field. Accordingly, it is vital for its curricula to maintain currency with the latest developments in the field.

Higher Education Commission (HEC) is investing substantial effort in improving and promoting higher education in the domain of curricula development and research. The following committees were constituted by HEC involving the respective expert faculty members both from public and private sectors throughout the country:

- National Curriculum Revision Committee-Computer Science (2009)-NCRC-CS
- National Curriculum Revision Committee-Software Engineering (2009)-NCRC-SE
- National Curriculum Revision Committee-Information Technology (2009)-NCRC-IT
- Joint National Curriculum Revision Committee (Computer Science, Software Engineering and Information Technology)-2009-JNCRC-CSSEIT

All committees held their preliminary meetings (except JNCRC) to establish the respective first draft of curriculum. The reports delivered by these committees were sent to the experts of international repute abroad for their evaluation and recommendations. Moreover, the same were also submitted to the various respective departments of universities for their review and feedback. Accordingly, final meetings were held to finalize the recommendations in their respective domains. All three committees developed a final report pertaining to the design, structure and courses details of BS, MS and PhD programs.

All three committees (NCRC-CS, NCRC-SE and NCRC-IT) worked independently in their respective domains through extensive interaction and consensus of national and international experts in the field. It is important to mention here that various delegates from international software industry including Microsoft and Oracle also participated in our meetings.

Subsequently, the following committee was constituted to develop a model to unify all the curricula and create systemic structures to maintain consistency of certain level in all the degree programs:

Meeting of Joint National Curriculum Revision Committee (Computer Science, Software Engineering and Information Technology)-2009

A two-day meeting of the Joint National Curriculum Revision Committee (Computer Science, Software Engineering and Information Technology)-2009 was held on June 12-13, 2009 at Higher Education Commission, Islamabad. All senior faculty members and experts in the domains of Computer Science, Software Engineering and Information Technology from both public and private sector institutions were invited nationwide to participate in the meeting.

The major objectives of the meeting include the following:

- **Unification among curricula of Computer Science, Software Engineering and Information Technology**

It is essential that consistent standards should to be maintained across all three curricula. To integrate the work of all three committees under the umbrella of Computing and to identify commonalities and differences among all three disciplines.

- **Curricula Revision Strategy**

The international scientific and professional bodies including Association of Computing Machinery (ACM), Institute of Electrical & Electronics Engineers (IEEE), and Joint ACM and IEEE Curriculum Task Force has already established Computing as an origin and basis for family of disciplines including Computer Science, Software Engineering and Information Technology.

The following participants attended the meeting and contributed significantly to establish a model to structure all degree programs on the basis of Computing in a systematic manner.

LIST OF EXPERT IN THE SUBJECT OF IT. CS & SE

Sr.	Name & Address
Federal / Rawalpindi	
1.	Prof. Dr. Aftab Ahmed Chairman National Computing Education Accreditation Council (NCEAC) Director Foundation University Institute of Management & Computer Sciences Near Lalazar Colony Rawalpindi.
2.	Prof. Dr. Aftab Maroof Professor National University of Computer & Emerging Sciences A.K Brohi Road H-11/4, Islamabad.
3.	Engr. Prof. Dr. Muhammad Yunus Javed, College of Electrical & Mechanical Engineering (EME), National University of Science and Technology, H-12, Islamabad.
4.	Dr. Muhammad Sher, Professor, Department of Computer Science, International Islamic University, H-10 Campus, Faculty Block-2, Islamabad
5.	Dr. Naveed Ikram, Associate Professor, Department of Computer Science, International Islamic University, H-10 Campus, Faculty Block-2, Islamabad
6.	Dr. Muhammad Yousaf, Associate Professor, Department of Computer Science & Engineering, Bahria University, Shangrila Road, Sector E-8, Islamabad

7.	Dr. Nazir A. Sangi, Professor, Department of Computer Science, Allama Iqbal Open University, Islamabad
8.	Dr. Saeed Bhatti, Professor, Deptt of Software Engineering Foundation University Medical College New Lalazar, Rawalpindi
9.	Dr. Farhana Shah, Convener NCRC IT Professor / Director, Quaid-i-Azam University, Institute of Information Technology, Islamabad
10.	Dr. Sharifullah Khan, Associate Professor, School of Electrical Engineering & Computer Sciences (SEECS), National University of Sciences and Technology, H-12, Islamabad.
11.	Dr. Aamer Nadeem Associate Professor M.A. Jinnah University (MAJU) Blue Area Islamabad.
12.	Prof. Dr. Abdul Qadir, Muhammad Ali Jinnah University (MAJU), Blue Area, Islamabad
13.	Mr. Munir Hussain Naveed, Assistant Professor, Department of Software Engineering, Fatima Jinnah Women University, The Mall, Rawalpindi
14.	Dr. Arshad Iqbal, Principal Engr. Deptt of Computer & Information Science P.O. PIEAS Nilore, Islamabad
15.	Dr. Shahid Nazir Bhatti Professor Deptt of Computer Science COMSATS Institute of Information Technology, 30, Sector H-8/1, Islamabad

16.	Mr. M. Imran Saeed Assistant Professor Department of Computer Science, International Islamic University, H-10 Campus, Faculty Block-2, Islamabad
17.	Dr. Mohammad Mahboob Yasin, Professor, Deptt of Computer Science, COMSATS Institute of Information Technology, 30, Sector H-8/1, Islamabad
18.	Prof. Dr. Amir Hayat, Head, Department of Computer Science, NUST School of Electrical & Computer Science, H-12, Islamabad.
19.	Dr. Arshad Ali Shahid, Professor, Deptt of Computer Science, National University of Computer & Engineering Science, A.K. Brohi Road, H-11/4, Islamabad
19	Dr. Muhammad Nadeem Khokhar Professor, Head of Department Computing Shaheed Zulfikar Ali Bhutto Institute of Science & Technology, Blue Area Islamabad.
20	Prof. Dr. Jamil Ahmed Dean Iqra University Islamabad Campus H-10, Islamabad.
21	Syed Afaq Husain Professor & Chairman, Deptt of Computer Science, Software Engineering & Information Technology, Air University E-9, Islamabad
Punjab	
1.	Prof. Dr. Muhammad Ali Maud Deptt of Computer Engg & I.T University of Engineering & Tech. Lahore.

2.	Dr. Iftikhar Hussain Shah, Professor, Deptt of Computer Science I.T. & Software Engineering Forman Christian College Lahore.
3.	Dr. Jerald Allan Kabell, Professor & Chairperson, Deptt of Computer & Information Technology, Forman Christian College, Ferozpur Road, Lahore 54600
4.	Dr. Sarmad Hussain, Professor & Head of Deptt of Computer Science, National University of Computer & Engineering Science, Lahore Campus Block B, Faisal Town, Lahore
5.	Dr. Shafay Shamail Chairman Deptt. Of Computer Science & Info. Tech LUMS, Lahore.
6.	Syed Mansoor Sarwar Principal Information Technolgy (PUCIT) Punjab University Lahore.
7.	Prof. Dr. Aftab Ahmed Malik, Professor, Department of Computer Science, B. Z. Univesity, Multan
Sindh	
1.	Dr. Imdad Ali Ismaili, Convener NCRC Computer Science Professor & Director, Institute of Information and Communication Technology, University of Sindh, Allama I. I. Kazi Campus, Jamshoro Sindh
2.	Dr. Sohail Asghar, Assistant Professor, Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science & Technology 90-Clifton Karachi

3.	Dr. Madad Ali Shah, Professor, Information Technology, IBA Sukkur Airport Road Sukkur.
4.	Dr. Ejaz Ahmed Professor, Deptt of Computer Science Institute of Business Management Korangi Creek Road Karachi-75190
5.	Prof. Dr. Aqil Burney, Convener NCRC in SE Chairman, Director Deptt. Of Computer Science , Info. Tech University of Karachi Karachi.
6.	Dr. Zubair A. Shaikh, Director, FAST National University of Computer & Emerging Sciences, Shah Latif Town, National Highway, Karachi
7.	Dr. Abdul Wahab Ansari, Professor, Institute of Information & Communication Technology, University of Sindh, Allama I. I. Kazi Campus, Jamshoro Sindh
8.	Dr. Qamar Uddin Khand, Professor, Deptt of Information Technology, IBA Sukkur Airport Road, Sukkur
9.	Dr. Akram Sheikh, Associate Professor, Department of CS and SE Engineering, Mehran University of Engineering & Technology, Jamshoro
10.	Prof. Dr. Najmi Ghani Haider, HOD, Department of Computer Sciences, Shaheed Zulfiqar Ali Bhutto Institute of Science & Technology (Szabist), 100 Clifton, Karachi

11.	Dr. Tahseen Ahmed Jilani Assistant Professor Department of Computer Science, University of Karachi, Karachi
12.	Mr. Muhammad Amir Associate Professor Deptt. Of Electronic Engg. Sir Syed University of Engg. & Tech Karachi.
13.	Prof. Dr. Abu Turab Alam College of Computer Science & Information System Korangi Creck Karachi.
NWFP	
1.	Dr. Muhammad Ali, Assistant Professor, Department of Information Technology & Computer Science, Institute of Management Sciences, 7/B-3, Phase-V, Hayatabad, Peshawar
2.	Dr. Muhammad Ajmal Bangash, Associate Professor, GIK Institute of Engineering Science and Technology, Topi – Swabi, NWFP
3.	Prof. Dr. M. Abid Khan, Department of Computer Science, University of Peshawar, Peshawar.
4.	Prof. Dr. Inayatullah Babar Chairman Deptt. Of Computer & IT NWFP, UET Peshawar.
5.	Prof. Dr. Asif Mehmood Gilani, GIK Institute of Engineering & Technology, Topi, District Swabi

Balochistan

1.	Prof. Dr. Abdul Hussain Shah Bukhari, Dean, Faculty of Information & Communication Technology (ICT), Balochistan University of Information Technology, Engineering & Management Sciences, Quetta
----	---

The following was sequence of presentations made during the meeting:

Sr. No	Topic	Presentation By
1	CS-Curriculum-2009	Dr. Imdad Ali Ismaili, Convener NCRC-CS Professor & Director, Institute of Information & Communication Technology, University of Sindh, Jamshoro
2	SE-Curriculum-2009	Dr. S. M. Aqil Burney, Convener NCRC-SE Meritorious Professor, Chairman and Director, Department of Computer Science, University of Karachi.
3	IT-Curriculum-2009	Prof. Dr Farhana Shah , Convener NCRC-IT Director, Institute of Information Technology Quaid-i-Azam University, Islamabad
4	Survey of the Latest Recommendation of ACM & IEEE Curriculum Task Force Regarding CS-Curriculum	Prof. Dr. Mansoor Sarwar Principal, Punjab University College of IT PU Old Campus, Lahore
5	Survey of the Latest Recommendation of ACM & IEEE Curriculum Task Force Regarding SE-Curriculum	Dr. Zubair A. Shaikh, Director, FAST-University of Computer and Emerging Sciences Shah Abdul Latif Town, National highway Karachi
6	Survey of the Latest Recommendation of ACM & IEEE Curriculum Task Force Regarding IT-Curriculum	Dr. Muhammad Abid, Head, Department of Computer Science, University of Peshawar
7	Survey of IT Industry Trends and What is Expected from Computing Programs in Pakistan?	Dr. M. Shoaib Khan, EME College NUST, Rawalpindi & Dr. Muid Mufti, ID Technologies, Islamabad

8	Software Industry Trends, Opportunities and Threats	Resource person from PSEB
9	Curricula Revision Strategy and Future Challenges	Prof. Dr. Aftab Ahmed Chairman National Computing Education Accreditation Council (NCEAC) Director Foundation University Institute of Management & Computer Sciences Near Lalazar Colony Rawalpindi.

Computing
Requirements for Bachelor Degree Programs
(Computer Science, Software Engineering, Information Technology)

Structure of BS Programs

	Name of Program	Computer Science	Software Engineering	Information Technology
#	Category	Credit Hours	Credit Hours	Credit Hours
1	Computing Courses	70	70	70
	Core Courses	43	43	43
	Supporting Areas	12	12	12
	General Education	15	15	15
2	Software Engineering Courses	48	48	48
	CS Core Courses	18	18	18
	CS Electives Courses	21	21	21
	CS Supporting Areas Courses (<i>Electives</i>)	9	9	9
3	University Electives	12	12	12
	Total Credit Hours	130	130	130

Structure of BS Program

Common Areas in all BS Programs of Computer Science, Software Engineering and Information Technology

Regarding

Computing, Supporting Area, General Education & University Elective Courses

A. Computing-Core Courses – 43 Credit Hours

#	Code	Pre-Req	Course Title	Credit hours	Proposed Semester
1		-	Introduction to Computing	4 (3-3)	1
2		-	Programming Fundamentals	4 (3-3)	1
3		1	Object Oriented Programming	3 (3-0)	2
4		-	Discrete Structures	3 (3-0)	2
5		3	Data Structure and Algorithms	3 (3-0)	3
6		3	Digital Logic and Design ¹	3 (3-0)	3
7		4	Operating Systems	4 (3-3)	4
8		4	Introduction to Database Systems	4 (3-3)	4
9		2	Introduction to Software Engineering ¹	3 (3-0)	4
10		6	Computer Communications and Networks ¹	3 (3-0)	6
11		8	Human Computer Interaction	3 (3-0)	7
12		-	Senior Design Project	6 (0-18)	7,8

1: Labs preferred in these courses. However, implementation details are left upon the concerned Institutes.

B. Supporting Area Courses - 12 Credit Hours

#	Code	Pre-Req	Course Title	Credit hours	Proposed Semester
13	MT		Calculus and Analytical Geometry	3 (3-0)	1
14	MT	-	Probability and Statistics	3 (3-0)	2
15	MT	-	Linear Algebra	3 (3-0)	4
16	PH	-	Electromagnetism	3 (3-0)	3

C. General Education Courses – 15 Credit Hours

#	Code	Pre-Req	Course Title	Credit hours	Proposed Semester
17	EG		English-I (Functional English)	3 (3-0)	1
18	EG		English-II (Technical and Report Writing)	3 (3-0)	2
19	EG	-	English-III (Communication Skills)	3 (3-0)	3
20	SS		Islamic and Pakistan Studies	3 (3-0)	1
21	SS	-	Professional Practices	3 (3-0)	8

University Elective Courses – 12 Credit Hours

(Not limited to the list below. Institutions may add more courses)

#	Code	Pre-Req	Course Title	Credit hours	Proposed Semester
1	MG	-	Financial Accounting	3(3, 0)	
2	MG	-	Financial Management	3 (3, 0)	4
3	MG	-	Human Resource Management	3 (3, 0)	5
4	MG	-	Marketing	3 (3, 0)	6
5	SS	-	Economics	3 (3, 0)	7
6	PS	-	Psychology	3 (3, 0)	6
7	SS	-	International Relations	3 (3, 0)	7
8	SS	-	Foreign/Regional Language (French, German, Sindhi, Punjabi, Urdu etc.)	3 (3, 0)	7-8
9	SS	-	Philosophy	3(3, 0)	6-8

COURSE CONTENTS

Computing – Core Courses (43 credit hours)

Course Name: **Introduction to Computing**

Course Structure: Lectures: 3, Labs: 1 **Credit Hours: 4**

Prerequisites: None

Objectives: This course focuses on a breadth-first coverage of computer science discipline, introducing computing environments, general application software, basic computing hardware, operating systems, desktop publishing, Internet, software applications and tools and computer usage concepts; Introducing Software engineering and Information technology within the broader domain of computing, Social issues of computing.

Course Outline: Number Systems, Binary numbers, Boolean logic, History computer system, basic machine organization, Von Neumann Architecture, Algorithm definition, design, and implementation, Programming paradigms and languages, Graphical programming, Overview of Software Engineering and Information Technology, Operating system, Compiler, Computer networks and internet, Computer graphics, AI, Social and legal issues.

Reference Material:

1. Computers: Information Technology in Perspective, 9/e by Larry Long and Nancy Long, Prentice Hall, 2002 / ISBN: 0130929891
2. *An Invitation to Computer Science*, Schneider and Gersting, Brooks/Cole Thomson Learning, 2000
3. *Computer Science: An overview of Computer Science*, Sherer,

Course Name: **Programming Fundamentals**

Course Structure: Lectures: 3, Labs: 1 **Credit Hours: 4**

Prerequisites: None

Objectives: The course is designed to familiarize students with the basic structured programming skills. It emphasizes upon problem analysis, algorithm designing, and programme development and testing.

Course Outline: Overview of computers and programming. Overview of language for e.g. C language C. Basics of structured and Modular programming. Basic Algorithms and problem solving, development of basic algorithms, analyzing problem, designing solution, testing designed solution. Fundamental programming constructs, translation of algorithms to programmes, data types, control structures, functions, arrays, records, files, testing programmes.

Reference Material:

1. Problem Solving and Program Design in C / 6E
Hanly & Koffman

Addison-Wesley | Published: 02/06/2009
ISBN-10: 0321535421 | ISBN-13: 9780321535429

2. C How to Program, 5/E

(Harvey & Paul) Deitel & Deitel, ISBN-10: 0132404168 ISBN-13:
9780132404167 Publisher: Prentice Hall Copyright: 2007

Course Name: Object Oriented Programming

Course Structure: Lectures: 2, Labs: 1 **Credit Hours:** 3

Prerequisites: Programming Fundamentals

Objectives: The course aims to focus on object-oriented concepts, analysis and software development.

Course Outline: Evolution of Object Oriented (OO) programming, OO concepts and principles, problem solving in OO paradigm, OO programme design process, classes, methods, objects and encapsulation; constructors and destructors, operator and function overloading, virtual functions, derived classes, inheritance and polymorphism. I/O and file processing, exception handling

Reference Material:

1. C++ How to Program, 6/E

(Harvey & Paul) Deitel & Deitel ISBN-10: 0136152503
ISBN-13: 9780136152507 Publisher: Prentice Hall

2. Java How to Program, 7/E

(Harvey & Paul) Deitel & Deitel ISBN-10: 0132222205 ISBN-13:
9780132222204 Publisher: Prentice Hall

Course Name: Discrete Structures

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives: Introduces the foundations of discrete mathematics as they apply to Computer Science, focusing on providing a solid theoretical foundation for further work. Further, this course aims to develop understanding and appreciation of the finite nature inherent in most Computer Science problems and structures through study of combinatorial reasoning, abstract algebra, iterative procedures, predicate calculus, tree and graph structures. In this course more emphasis shall be given to statistical and probabilistic formulation with respect to computing aspects.

Course Outline: Introduction to logic and proofs: Direct proofs; proof by contradiction, Sets, Combinatorics, Sequences, Formal logic, Propositional and predicate calculus, Methods of Proof, Mathematical Induction and Recursion, loop invariants, Relations and functions, Pigeonhole principle, Trees and Graphs, Elementary number theory, Optimization and matching. Fundamental structures: Functions; relations (more specifically recursions); pigeonhole principle; cardinality and countability, probabilistic methods.

Reference Material:

1. Kenneth H. Rosen, *Discrete Mathematics and Its Applications*, 6TH edition, 2006, Mcgraw Hill Book Co.

2. Richard Johnsonbaugh, *Discrete Mathematics*, 7TH edition, 2008, Prentice Hall Publishers.
3. Kolman, Busby & Ross, *Discrete Mathematical Structures*, 4th edition, 2000, Prentice-Hall Publishers.
4. Ralph P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, Addison-Wesley Pub. Co., 1985.

Course Name: Operating Systems

Course Structure: Lectures: 3, Labs: 1 **Credit Hours: 4**

Prerequisites: None

Objectives: To help students gain a general understanding of the principles and concepts governing the functions of operating systems and acquaint students with the layered approach that makes design, implementation and operation of the complex OS possible.

Course Outline: History and Goals, Evolution of multi-user systems, Process and CPU management, Multithreading, Kernel and User Modes, Protection, Problems of cooperative processes, Synchronization, Deadlocks, Memory management and virtual memory, Relocation, External Fragmentation, Paging and Demand Paging, Secondary storage, Security and Protection, File systems, I/O systems, Introduction to distributed operating systems. Scheduling and dispatch, Introduction to concurrency.

Lab assignments involving different single and multithreaded OS algorithms.

Reference Material:

1. *Applied Operating Systems Concepts*, 7th Edition, Silberschatz A., Peterson, J.L., & Galvin P.C. 2004.
2. *Modern Operating Systems*, 3rd Edition, Tanenmaum A.S., 2008.

Course Name: Database Systems

Course Structure: Lectures: 3, Labs: 1 **Credit Hours: 4**

Prerequisites: Data Structures and Algorithms

Objectives: The course aims to introduce basic database concepts, different data models, data storage and retrieval techniques and database design techniques. The course primarily focuses on relational data model and DBMS concepts.

Course Outline: Basic database concepts; Entity Relationship modelling, Relational data model and algebra, Structured Query language; RDBMS; Database design, functional dependencies and normal forms; Transaction processing and optimization concepts; concurrency control and recovery techniques; Database security and authorization. Small Group Project implementing a database. Physical database design: Storage and file structure; indexed files; b-trees; files with dense index; files with variable length records; database efficiency and tuning.

Reference Material:

1. *Database Systems 8E*, C.J.Date, Addison Wesley Pub. Co. (2004).
2. *Database Systems: A Practical Approach to Design, Implementation and Management 5E*, R.Connolly and P.Begg, Addison-Wesley Pub. Co (2009).
3. *Fundamentals of Database Systems, 5/E*, Elmasri and Navathe, Addison-Wesley, ISBN: 0-201-74153-9.

Course Name: Introduction to Software Engineering

Course Structure: Lectures: 2, Labs: 1 **Credit Hours: 3**

Prerequisites: Object Oriented Paradigm/Programming

Objectives: To study various software development models and phases of software development life cycle. The concepts of project management, change control, process management, software development and testing are introduced through hands-on Team Projects.

Course Outline: Introduction to Computer-based System Engineering; Project Management; Software Specification; Requirements Engineering, System Modelling; Requirements Specifications; Software Prototyping; Software Design: Architectural Design, Object-Oriented Design, UML modelling, Function-Oriented Design, User Interface Design; Quality Assurance; Processes & Configuration Management; Introduction to advanced issues: Reusability, Patterns; Assignments and projects on various stages and deliverables of SDLC.

Reference Material:

1. *Software Engineering 8E by Sommerville Addison Wesley, 2006*
2. *Software Engineering: A Practitioner's Approach 7E*, Roger Pressman, McGraw-Hill, 2009

Course Name: Computer Communication and Networks

Course Structure: Lectures: 2, Labs: 1 **Credit Hours: 3**

Prerequisites: None

Objectives: To introduce students to the concept of computer communication. Analogue & digital transmission. Network Layers, Network models (OSI, TCP/IP) and Protocol Standards. Emphasis is given on the understanding of modern network concepts.

Course Outline: Analogue and digital Transmission, Noise, Media, Encoding, Asynchronous and Synchronous transmission, Protocol design issues. Network system architectures (OSI, TCP/IP), Error Control, Flow Control, Data Link Protocols (HDLC, PPP). Local Area Networks and MAC Layer protocols (Ethernet, Token ring), Multiplexing, Switched and IP Networks, Inter-networking, Routing, Bridging, Transport layer protocols TCP/IP, UDP. Network security issues. Programming exercises, labs or projects involving implementation of protocols at different layers.

Reference Material:

1. Introduction to Computer Networks /4, A. S. Tanenbaum, Prentice Hall 2003
2. Computer Networks and Internets, 5/E, 2008
Douglas E. Comer, Purdue University ISBN-10: 0136061273 ISBN-13:
9780136061274 Publisher: Prentice Hall
3. Data and Computer Communications By William Stallings Published by
Macmillan Pub. Co., 8th Edition 2006

Course Name: Human Computer Interaction

Course Structure: Lectures: 2, Labs:1 **Credit Hours: 3**

Prerequisites: Data Structures and Algorithms

Objectives: This course introduces the human issues of usability and its importance. It considers the implications of human understanding on the usability of computer systems and the importance of understanding the context of use. It describes guidelines for use of different media and interface styles. Topics include Usability Design principals, standards and models, evaluation techniques. Groupware, pervasive and ubiquitous applications.

Course Outlines: The Human, Computer and Interaction, Usability paradigm and principles, Introduction to design basics, HCI in software process, Design rules, prototyping, evaluation techniques, task analysis, Universal design and User support and Computer Supported Cooperative Work. Introduction to specialized topics such as Groupware, pervasive and ubiquitous applications.

Resources:

1. Human-Computer Interaction, 3/E **Alan Dix**, *Computing Dept, Lancaster University*
Janet E. Finlay, *Leeds Metropolitan University*, **Gregory D. Abowd**, *Georgia Institute of Technology*, **Russell Beale**, *University of Birmingham* ISBN-10:
0130461091
ISBN-13: 9780130461094 Publisher: Prentice Hall
2. Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4/E **Ben Shneiderman**, *University of Maryland* **Catherine Plaisant**, *University of Maryland* ISBN-10: 0321197860 ISBN-13:
9780321197863 Publisher: Addison-Wesley

Course Name: Senior Design Project

Course Structure: Lectures: 0, Labs: 6 **Credit Hours: 6**

Prerequisites: Introduction to Software Development, Data Base Systems, Computer Architecture

Objectives: The software project involves research, conceive, plan and develop a real and substantial project related to computer science. It provides an opportunity to the students to crystallize their acquired professional competence in the form of a demonstrable software product. Make oral and written project presentations.

Resources:

1. *Software Project Management in Practice* by Jalote, Pankaj.

Computing – Supporting Courses (12 credit hours)

Course Name: **Calculus and Analytic Geometry**

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives: To provide foundation and basic ground for calculus and analytical geometry background.

Course Outline: Complex Numbers, DeMoivre's Theorem and its Applications, Simple Cartesian Curves, Functions and Graphs, Symmetrical Properties, Curve Tracing, Limit and Continuity, Differentiation of Functions. Derivative as Slope of Tangent to a Curve and as Rate of Change, Application to Tangent and Normal, Linearization, Maxima/Minima and Point of Inflexion, Taylor and Maclaurin Expansions and their convergence. Integral as Anti-derivative, Indefinite Integration of Simple Functions. Methods of Integration: Integration by Substitution, by Parts, and by Partial Fractions, Definite Integral as Limit of a Sum, Application to Area, Arc Length, Volume and Surface of Revolution.

Reference Material:

1. Swokowski, Olinick and Pence, *Calculus and Analytical Geometry*, 6th edition, 1994, Brooks/Cole Publishers.
2. Howard Anton, *Calculus*, 7th edition. 2002, John Wiley and Sons (WIE).
3. William E. Boyce Richard C. Diprima, *Calculus*, John Wiley & Sons, ISBN: 0471093335.
4. Thomas Finny, *Calculus and Analytical Geometry*, 10th edition, John Wiley and Sons.
5. Erwin Kreyzig, *Advanced Engineering Mathematics*, 7th edition, 1993, John Wiley & Sons Inc.

Course Name: **Probability and Statistics**

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives: To introduce the concepts of data analysis, presentation, counting techniques, probability and decision making.

Course Outline: Introduction to Statistics, Descriptive Statistics, Statistics in decision making, Graphical representation of Data Stem-and Lead plot, Box-Cox plots, measures of central tendencies and dispersion, moments of frequency distribution; Counting techniques, introduction to probability, sample space, events, laws of probability, Conditional probability and Baye's theorem with application to random variable (Discrete and continuous) Binomial, Poisson, Geometric, Negative Binomial Distributions; Exponential Gamma and Normal distributions. Regression and Correlation, Estimation and testing of hypotheses, use of elementary statistical packages for explanatory Data analysis.

Reference Material:

1. Ronald Walpole, Myers, Myers, Ye, "Probability & Statistics for Engineers & Scientists", 8th edition, 2008, **Prentice Hall** Publisher.

2. Lay L. Devore, Probability and Statistics for Engineering and the Sciences, 2003, Duxbury Publishers.
3. G. Cowan, *Statistical Data Analysis*, 1998, Clarendon, Oxford.

Course Name: Linear Algebra

Course Structure: Lectures: 3, Labs: 0 **Credit Hours: 3**

Prerequisites: None

Objectives: To provide fundamentals of solution for system of linear equations, operations on system of equations, matrix properties, solutions and study of their properties.

Course Outline: Vectors, Vector Spaces, Matrices & Determinants, Cofactor and Inverse, Rank, Linear Independence, Solution of system of Linear systems, Positive Definite matrix, Linear Transformations, Operations on matrices, Inner products, orthogonality and least squares, Eigenvalue & Eigenvectors. Applications to Systems of Equations and to Geometry, Singular Value Decomposition.

Reference Material:

1. Bernard Kolman, David Hill, Elementary Linear Algebra with Applications, 9th edition, Prentice Hall PTR, 2007.
2. Gilbert Strang, Strang, Brett Coonley, Andy Bulman-Fleming, Andrew Bulman-Fleming, Strang's Linear Algebra And Its Applications, 4th edition, Brooks/Cole, 2005
3. Howard Anton, Chris Rorres, Elementary Linear Algebra: Applications Version, 9th edition, Wiley, 2005.
4. David C. Lay, Linear Algebra and Its Applications, 2nd edition, Addison-Wesley, 2000.

Course Name: Electromagnetism

Course Structure: Lectures: 2, Labs: 1 **Credit Hours: 3**

Prerequisites: None

Objectives: Introduction of Electronics

Course Outline: Fundamentals of Semiconductor physics: Band theory, semiconductors (intrinsic and extrinsic), pn junction, pn junctions as a rectifier, clipper and clamper circuits, zener diode and voltage regulator, LED and LCD etc., *Transistors:* Bipolar Junction transistors, BJT biasing circuits, Q-point, BJT as a switch, BJT amplifiers, classes of amplifiers, power amplifiers, Metal oxide transistors, nMOS, pMOS and CMOS inverters circuits. Introduction to A/D and D/A conversion circuits.

Reference Material:

1. Freedman and Young, *University Physics*, (10th and higher editions).
2. Resnick, Halliday and Krane, *College Physics* (6th and higher edition).

Computing – General Education Courses (15 credit hours)

Course Name: **English-I (Functional English)**

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Prerequisites: None

Annexure - A

Course Name: **English – II (Technical and Report Writing)**

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Prerequisites: None

Annexure - B

Course Name: **English – III (Communication Skills)**

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Prerequisites: None

Annexure - C

Course Name: **Islamic & Pakistan Studies**

Course Structure: Lectures: 3, Labs: 0 **Credit Hours:** 3

Prerequisites: None

Annexure – D&E

Course Name: **Professional Practice**

Course Structure: Lectures:3, Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives: A Computing graduate as professional has some responsibilities with respect to the society. This course develops student understanding about historical, social, economic, ethical, and professional issues related to the discipline of Computing. It identifies key sources for information and opinion about professionalism and ethics. Students analyze, evaluate, and assess ethical and professional computing case studies.

Course Outline: Historical, social, and economic context of Computing (software engineering, Computer Science, Information Technology); Definitions of Computing (software engineering, Computer Science, Information Technology) subject areas and professional activities; professional societies; professional ethics; professional competency and life-long learning; uses, misuses, and risks of software; information security and privacy; business practices and the economics of software; intellectual property and software law (cyber law); social responsibilities, software related contracts, Software house organization

Resources:

1. *Professional Issues in Software Engineering*, M.F. Bott et al.

National Curriculum Revision Committee – Computer Science (2009)

A three day final meeting of the National Curriculum Revision Committee of Computer Science, pertaining to revising the curriculum for Computer Science degree programs developed in 2004 was held from February 17th to 19th, 2009 at HEC regional Centre, Karachi. The aims and objectives of the meeting were to discuss the deliberations and finalize the curriculum drafted by the committee of the last meeting held from 28th to 30th August, 2008.

The following attended the meeting;

1	Dr. Imdad Ali Ismaili, Professor & Director, Institute of Information & Communication Technology, University of Sindh, Jamshoro	Convener
2	Mr. Badar Sami, Associate Professor, Department of Computer Science, University of Karachi, Karachi	Secretary
3	Dr. Zubair A. Sheikh, Director, Department of Computer & Science, National University of Computer & Engineering Science, Karachi Campus, Shah Latif Town, National Highway, Karachi	Member
4	Prof. Dr. Aftab Ahmed Malik, Professor, Department of Computer Science, B. Z. Univesity, Multan	Member
5	Dr. Tahseen Ahmed Jilani, Department of Computer Science, University of Karachi, Karachi	Member
6	Dr. Mohammad Mahboob Yasin, Professor, Department of Computer Science, COMSATS Institute of Information Technology, 30, Sector H-8/1, Islamabad	Member
7	Dr. Amir Hayat, Head, Department of Computing, NUST School of Electrical Engineering & Computer H-12, Islamabad	Member

8	Dr. M. Imran Saeed Assistant Projector, Department of Computer Science, International Islamic University, H-10 Campus, Faculty Block-2, Islamabad	Member
9	S. Zafar Nasir, Professor FAST National University, Karachi	Member
10	Prof. Dr. Asif Mehmood Gilani, GIK Institute of Engineering & Technology, Topi, District Swabi	Member
11	Prof. Dr. Abu Turab Alam, College of Computer Science & Information System, Karachi	Member
12	Dr. Nazir A. Sangi, Professor, Department of Computer Science, Allama Iqbal Open University, Islamabad	Member
13	Dr. Madad Ali Shah, Professor, Deptt of Information Technology, Institute of Business Administration, Airport Road Sukkur	Member
14	Dr. Asadullah Shah, Dean, Faculty of Computer Management Sciences, Isra University, Hyderabad	Member
15	Dr. Arshad Ali Shahid, Professor, Deptt of Computer Science, National University of Computer & Engineering A.K. Brohi Road, H-11/4, Islamabad	Member
17	Syed Afaq Hussain, Professor & Chairman, Department of Computer Science, Software Engineering & Information Technology, Air University E-9, Islamabad	Member
18	Dr. Iftikhar Hussain Shah, Professor, Deptt of Computer Science & IT, F.C College, Lahore	Member

19	Mr. Farrukh Amin, Assistant Professor (CS), Institute of Business Management, Korangi Creek, Karachi	Member
20	Meritorious Prof. Dr. S. M. Aqil Burney Chairman Department of Computer Science, University of Karachi	Member
21	Dr. Sh. M. Wahabuddin Usmani, Associate Professor, Department of CS & IT, NED University of Engineering & Technology, University Road, Karachi	Member

The proceedings of the meeting started with the recitation from the holy Quran by Mr. Tahir Ali Shah and welcome address by Mr. Rafiq Rai (Director – HEC Karachi region). The house unanimously nominated and elected Dr. Imdad Ali Ismaili as Convener and Mr. Badar Sami as secretary of the committee.

The Convener of the meeting Prof. Dr. Imdad Ali Ismaili informed the participants that the comprehensive report of previous meeting held on 28th to 30th August 2008 has been already circulated among all the members of the committee, universities and institutions to have feedback from them.

The following programs were discussed by the participants of the meeting

- 1) BS (Computer Science)
- 2) MS (Computer Science)

After detailed revision of the minutes of the previous meeting held on August 28th to 30th 2008, following recommendations were made for BS (Computer Science) program to achieve the desired objectives;

Revision recommendations regarding BS (Computer Science) program

- 1) Digital Logic Design should be a separate course of 3 credit hours instead of being part of “Digital Logic & Computer Architecture” course and should be included in the “Computing – Core”. “Computer Architecture” should be included in the “Computer Science – Core” as a full 3 credit hour course in place of “System Programming” and “System Programming” may be placed in “Computer Science – Elective” courses .
- 2) Digital Computer Logic should not have any pre-requisite

- 3) The credit hours for each course should be written using standard notations.
- 4) Discrete Structures be moved from “Computing Core” to “Computer Science – Required Supporting courses”
- 5) Contents of “Electromagnetism” should be included in the contents of “Basic Electronics”
- 6) Numerical Computing should be moved from “Computer Science – Electives” to “Computer Science – Core” and contents may be updated to include Symbolic Computing related topics.
- 7) Each “Computer Science – Elective” area/group should comprise of at least four courses. Since the areas of “Software Engineering”, “Multimedia” have less than 4 courses so more courses be included in those areas/groups.
- 8) New areas/ groups may be included in “Computer Science – Electives” (like Entrepreneurship, Multimedia Computing etc.)
- 9) The committee members also emphasized the need of revising the contents of the courses of the schemes for BS and MS(Computer Science) and advised to update books and reference material.

BS (Computer Science)

Curriculum for BS (Computer Science) Program:

The same basis used in the last meeting held in 2004 were taken to revise the Curriculum of Computer Science.

1. Objectives/Goals
2. Strategies
3. Fast Changing Disciplines
4. Emerging Technologies
5. International Standards
6. Industrial Challenges
7. Possible program design structure

Almost all the members of the committee unanimously approved the proposed objectives of the program, program structure, general recommendation regarding the update and revise of the curriculum.

The structure and other details of the program proposed by the committee were designed inline to the recommendations of various leading bodies continuously in the quest to designing the educational programs of Computer Science and related disciplines. These bodies include IEEE and ACM. Latest reports and recommendations of “**Computer Science Curriculum 2008: Interim Revision of CS2001 report**” by **Interim Review Task Force of ACM and IEEE Computer Society** were mainly considered.

Many changes were recommended in various sections of the curricula developed by this Committee in the last meeting held in August 2008. The Committee finally agreed to the curriculum model presented in the following table.

#	Category	Credit Hrs
1	Computing courses	
	Computing – Core courses	43
	Computing – Supporting areas	12
	Computing – General Education	15
		70
2	Computer Science courses	
	CS – Core courses	18
	CS – Electives	21
	CS – Supporting Area	9
		48
3	University Electives	12
	Total credit hours	130

A complete detail of BS programme involving objectives, structure, distribution of credits among various components of programme are discussed in the following pages.

Objectives

Recent developments in computer hardware, software and communication technologies have offered new exciting opportunities and challenges for creation of innovative learning environments for Computer Science and its curricula design. One of the key elements here is to prepare the graduates for the future. The challenge of getting all newly emerging technologies incorporated in to the curriculum is becoming pivotal for the effectiveness of curricula. There is a need for curricula structures that are really able to grow as we put new demands on them. The curriculum is required to provide integration of all components and the foundations that allow accessing all of the new knowledge and technology to fulfil the vision of future.

The basic intention of an academic programme in Computer Science is to develop the student's critical professional thinking and intuition. The curriculum must be structured to provide a balanced mixture of learning experiences to make the graduate capable of sound professional decisions. As a result the graduate should be able to assume responsible positions in business, government, and education at the research, development, and planning levels. The programme should also provide an excellent foundation for further formal learning and training. The Computer Science curriculum is expected to provide environments to put into practice, the principles and techniques learnt during the course of implementation of academic programme.

The following summarizes some key characteristics for consideration as a basis of a successful academic programme in Computer Science:

1. The programme should provide a broad understanding of the field via introducing concepts, theory, and techniques.
2. Intensive education/training in focused areas of Computer Science is desirable.
3. The programme may encourage students to develop and use abstract models in addition to apply respective technology in practical situations.
4. Computer Science graduates require special communication skills both orally and in writing. They must be able to produce well-organized reports, which clearly delineate objectives, methods of solution, results, and conclusions for a complex task.
5. The programme should provide formal foundations for higher learning.
6. The programme should be dynamic and flexible enough to maintain currency with the latest scientific and technological developments in the field.
7. The programme should provide professional orientation to prepare students for industry.

Programme Structure

The structure of a BS programme in Computer Science is proposed to meet the needs of students with formal computing experience and with established relevant skills. The students are expected to learn theoretical and practical understanding of the entire field of Computer Science.

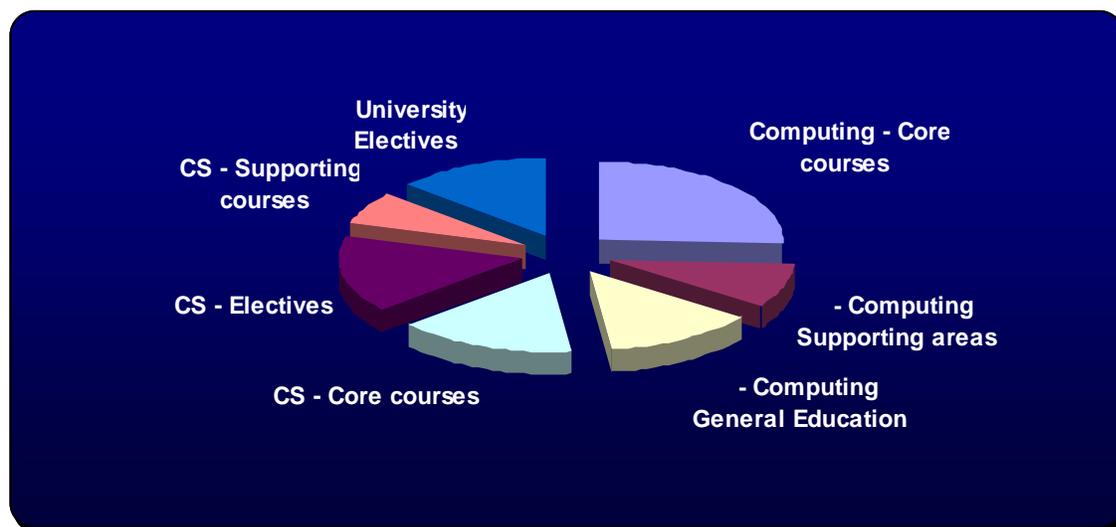
The proposed structure is dynamic and provides basis for various options including Breadth-Based, Depth-Based, and Integrated Breadth & Depth-Based specializations. Student may choose a particular option, which is most appropriate to their planned future career. The following are relevant details:

Minimum credit hours shall be 133 for BS (Computer Science) programme including computing related courses.

The programme shall comprise 8 semesters spread over 4 years with two semesters a year. The major area of specialization shall be incorporated in the structure. Each major area shall comprise of 4-6 courses.

The following is distribution of total credit hours.

Course Group	Credit hour	Percentage
Computing - Core courses	43	33%
Computing - Supporting areas	12	9%
Computing - General Education	15	12%
CS - Core courses	18	14%
CS – Electives	21	16%
CS - Supporting courses	9	7%
University Electives	12	14%



Some clusters regarding Computer Science Electives are listed below:

- a) Networking
- b) Information Management
- c) Intelligent Systems
- d) Graphics & Visualization
- e) Software Engineering
- f) Web Engineering
- g) E-Commerce
- h) Multimedia
- i) Distributed Computing
- j) Security
- k) Languages and Translators
- l) Computer Architecture
- m) Systems Software
- n) Scientific Computing
- o) Soft Computing

University Electives

It was unanimously recommended that 18 credit hours shall require to be taken from the list of general elective courses. The university may add any number of courses to the general elective courses preferably other than Computer Science courses.

Eligibility Criteria

The eligibility criteria of the draft curriculum by the last meeting were opened for discussion in the House. It was thoroughly discussed by considering all input streams of BS (Computer Science). The House unanimously recommended the eligibility criteria for admission to BS (Computer Science) as given:

The candidates must have intermediate or equivalent qualification. However, the university shall define their selection criteria.

General Recommendation Regarding Implementation of Programme

Faculty level and orientation is vital for the successful implementation. It is strongly recommended that the BS programme should be only implemented via experienced computer science faculty having formal education in Computer Science.

The access to state of the art computing and information technology is essential for creation of innovative learning environments. Professional areas of specialization such as computer graphics, multimedia systems, computer networking and virtual reality or design automation require very special and dedicated computing facilities. Dedicated computing facilities are essential for hands-on experience. Variety of programming languages systems and operating systems must be available.

Besides faculty and computing facilities, substantial library resources are important to support a rigorous graduate programme in information technology. Students

should have access to digital libraries and knowledge resources via Internet technologies.

Related IT Curriculum Efforts

There are various major curriculum efforts that relate to the Computer Science curricula:

- a) The IFIP (International Federation of Information Processing) Curriculum Reports
- b) The DPMA (Data Processing Management Association) Computer Systems Proposal
- c) The ACM (Association of Computing Machinery) Curriculum Task Force-Curriculum 2001
- d) The ITAA (Information Technology Association of America) Report on IT Workforce Study

Scheme of Studies for Bachelor Degree Computer Science Program

Computing courses

Computing - Core Courses (34 Credit Hours)					
#	Code	Pre-Req	Course Title	Credit hours	Proposed Semester
1	CS	-	Programming Fundamentals	3 (2, 1)	1
2	CS	1	Object Oriented Programming	3 (2, 1)	2
3	CS	2	Data Structure and Algorithms	3 (2, 1)	3
4	CS	21	Digital Logic Design	3 (2, 1)	3
5	CS	-	Operating Systems	3 (2, 1)	5
6	CS	3	Database Systems	3 (2, 1)	5
7	CS	2	Introduction to Software Development	3 (3, 0)	6
8	CS	-	Computer Communications and Networks	3 (2, 1)	6
9	CS	6, 7, 26	Final year Project	6	7, 8
10	CS	3	Human Computer Interaction	3 (2, 1)	7

Computing – Supporting Courses (12 Credit Hours)					
#	Code	Pre-Req	Course Title	Credit hours	Proposed Semester
11	MT	-	Calculus and Analytical Geometry	3 (3, 0)	1
12	MT	-	Probability and Statistics	3 (3, 0)	3
13	MT	-	Linear Algebra	3 (3, 0)	2
14	EL	-	Basic Electronics	3 (2, 1)	1

Computing – General Education Courses (18 Credit Hours)					
#	Code	Pre-Req	Course Title	Credit hours	Proposed Semester
16	EG	-	English-I (Functional English)	3 (3, 0)	1
17	EG	-	English-II (Technical and Report Writing)	3 (3, 0)	2
18	EG	-	English-III (Communication Skills)	3 (3, 0)	3
19	PK	-	Islamic and Pakistan Studies	3 (3, 0)	2
20	SS	--	Professional Practices	3 (3, 0)	7
21	CS	-	Introduction to Information and Communication Technologies	3 (2, 1)	1

Computer Science courses

Computer Science – Core Courses (18 Credit Hours)					
#	Code	Pre-Req	Course Title	Credit Hours	Proposed Semester
22	CS	4	Computer Organization and Assembly Language	3 (2, 1)	4
23	CS	29	Theory of Automata & Formal Languages	3 (3, 0)	5
24	CS	3, 29	Design and Analysis of Algorithms	3 (3, 0)	7
25	CS	29	Artificial Intelligence	3 (2, 1)	8
26	CS		Computer Architecture	3 (2, 1)	6
27	CS	23	Compiler Construction	3 (2, 1)	7

Computer Science – Supporting Courses (9 Credit Hours)					
#	Code	Pre-Req	Course Title	Credit Hours	Proposed Semester
30	ST	11	Multivariate Calculus	3 (3, 0)	4
31	ST	11	Differential Equations	3 (3, 0)	5
	CS	11	Numerical Computing	3 (3,0)	7

Computer Science – Elective Courses (21 Credit Hours) –(Not limited to the list below)					
#	Code	Area	Course Title	Credit hours	Proposed Semester
32	CS	CG V ¹	Computer Graphics	3 (2, 1)	6
33	CS	CG V ¹	Digital Image Processing	3(2, 1)	
34	CS	CG V ¹	Digital Signal Processing		
35	CS	CG V ¹	Computer Vision		
36	CS	SE ²	Software Engineering	3 (3, 0)	5
37	CS	SE ²	Advance Software Engineering		
38	CS	LT ³	Principles of Programming Languages	3 (2, 1)	4
39	CS	CC N ⁴	Data Communication	3 (3, 0)	6
40	CS	CC N ⁴	Distributed Computing	3 (2, 3)	6
41	CS	CC N ⁴	Data and Network Security	3(3, 0)	7

42	CS	CC N ⁴	Wireless Networks	3(2, 3)	
43	CS	CC N ⁴	Telecommunication Systems	3 (2, 1)	
44					
45					
46	CS	CA O ⁶	Microprocessor Interfacing	3 (2, 1)	7
47	CS	WE 7	Web Engineering	3 (2, 1)	5
48	CS	SS ⁸	System Programming	3 (2, 1)	7
49	CS	IM ⁹	Distributed Database Systems	3 (2, 1)	7
50	CS	IM ⁹	Data Warehousing	3(2, 1)	6-7
51	CS	SIC 10	Numerical and Symbolic Computing		
52	CS	SIC 10	Operations Research		
53	CS	SIC 10	Simulation and modelling		
54	CS	SO C ¹¹	Expert Systems		
55	CS	SO C ¹¹	Artificial Neural Network		
56	CS	SO C ¹¹	Fuzzy Logic		

- 1 – CGV = **Computer Graphics and Visualization**
 2 – SE = **Software Engineering**
 3 – LT = **Languages and Translators**
 4 – CCN = **Computer Communication Networks**
 7 – WE = **Web Engineering**
 8 – SS = **Systems Software**
 9 – IM = **Information Management**
 10 – SIC = **Scientific Computing**
 11 – SOC = **Soft Computing**

University Elective courses

Computer Science – University Elective Courses (18 Credit Hours) –(Not limited to the list below)					
#	Code	Pre-Req	Course Title	Credit Hours	Proposed Semester
101	MG	-	Financial Accounting	3(3, 0)	
102	MG	-	Financial Management	3 (3, 0)	4
103	MG	-	Human Resource Management	3 (3, 0)	5
104	MG	-	Marketing	3 (3, 0)	6
105	SS	-	Economics	3 (3, 0)	7
106	PS	-	Psychology	3 (3, 0)	6
107	SS	-	International Relations	3 (3, 0)	7
108	SS	-	Foreign/Regional Language (French, German, Sindhi, Punjabi, Urdu etc.)	3 (3, 0)	7-8
109	SS	-	Philosophy	3(3, 0)	6-8

Sample Scheme of Study for BS (CS) 4–year Program (8 Semesters) (130 Credit Hours)

Semester-wise 4-Year Plan

Semester 1	Cr. Hrs.	Semester 2	Cr. Hrs.
Introduction to Computing	4	Discrete Structures	3
Programming Fundamentals	4	Object Oriented Programming	3
Calculus and Analytical Geometry	3	Multivariable Calculus	3
Pakistan Studies and Islamic Studies	3	Probability and Statistics	3
English-I (Functional English)	3	English-II (Technical and Report Writing)	3
	17		15
 Semester 3	 Cr. Hrs.	 Semester 4	 Cr. Hrs.
Digital Logic and Design	3	Operating Systems	4
Data Structures and Algorithms	3	Differential Equations	3
Linear Algebra	3	Introduction to Database Systems	4
English-III (Communication Skills)	3	Introduction to Software Engineering	3
Electromagnetism	3	Computer Organization and Assembly Language	3
University Elective I	3		
	18		17

Semester 5	Cr. Hrs.	Semester 6	Cr. Hrs.
Computer Communication and Networks	3	Compiler Construction	3
Theory of Automata & Formal Languages	3	CS Elective I	3
Computer Architecture	3	Numerical Computing	3
Human Computer Interaction	3	Design and Analysis of Algorithms	3
University Elective II	3	CS Elective II	3
University Elective III	3	University Elective IV	3
	18		18
	Cr. Hrs.	Semester 8	Cr. Hrs.
Semester 7		Software Design Project II	3
Software Design Project I	3	CS Elective V	3
Professional Practices	3	CS Elective VI	3
CS Elective III	3	CS Elective VII	3
CS Elective IV	3		
Artificial Intelligence	3		
	15		12

COURSE CONTENTS

For BS Computer Science

Computer Science – Core Courses (18 credit hours)

Course Name: Computer Organization and Assembly Language

Course Structure: Lectures: 2, Labs: 1 **Credit Hours: 3**

Prerequisites: Digital Logic Design

Objectives: The main objective of this course is to introduce the organization of computer systems and usage of assembly language for optimization and control. Emphasis should be given to expose the low-level logic employed for problem solving while using assembly language as a tool. At the end of the course the students should be capable of writing moderately complex assembly language subroutines and interfacing them to any high level language.

Course Outline: Microprocessor Bus Structure: Addressing, Data and Control, Memory Organization and Structure (Segmented and Linear Models), Introduction to Registers and Flags, Data Movement, Arithmetic and Logic, Programme Control, Subroutines, Stack and its operation, Peripheral Control Interrupts, Interfacing with high level languages, Real-time application.

Objectives and Perspectives of Assembly Language, Addressing Modes, Introduction to the Assembler and Debugger, Manipulate and translate machine and assembly code, Describe actions inside the processing chip, Discuss operations performed by an instruction set, Write a fully documented program, Using an assembler of choice.

Reference Material:

1. Stallings, "Computer Organization & Architecture", 7th ed, Prentice HALL, 2006.
2. Irvine, Assembly Language for Intel-based Computers, 5th ed, Prentice Hall, 2007.
3. Computer Organization and Design, The Hardware/Software Interface, 4th ed, by David A. Patterson and John L. Hennessy, 2008. Elsevier Publishers.

Course Name: Theory of Automata and Formal languages

Course Structure: Lectures: 3 Labs: 0 **Credit Hours: 3**

Prerequisites: Discrete Structures

Objectives: The course aims to develop an appreciation of the theoretical foundations of computer science through study of mathematical & abstract models of computers and the theory of formal languages. *Theory of formal languages* and use of various abstract machines as 'recognizers' and parsing will be studied for identifying/validating the synthetic characteristics of programming languages. Some of the abstract machines shall also study as 'Transducers'.

Course Outline: *Finite State Models:* Language definitions preliminaries, Regular expressions/Regular languages, Finite automata (FAs), Transition graphs (TGs), NFAs, Kleene's theorem, Transducers (automata with output),

Pumping lemma and non regular language *Grammars and PDA*: Context free grammars, Derivations, derivation trees and ambiguity, Simplifying CFLs , Normal form grammars and parsing, Decidability, Chomsky's hierarchy of grammars *Turing Machines Theory*: Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Context sensitive Grammars, Defining Computers by TMs.

Text Books/Reference Books:

1. An Introduction to Formal Languages and Automata, By Peter Linz, 4th edition, Jones & Bartlett Publishers, 2006
2. Theory of Automata, Formal Languages and Computation, By S. P. Eugene, Kavier, 2005, New Age Publishers, ISBN (10): 81-224-2334-5, ISBN (13) : 978-81-224-2334-1.
3. John Hopcroft and Jeffrey Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd edition, 2001, Addison-Wesley.
4. Introduction to Languages and the Theory of Computation, By John C. Martin 3rd edition, 2002, McGraw-Hill Professional.

Course Name: Design and Analysis of Algorithms

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: Discrete Structure, Data Structures and Algorithms

Objectives: Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis on the structure, complexity, and efficiency of algorithms.

Course Outline: Introduction; Asymptotic notations; Recursion and recurrence relations; Divide-and-conquer approach; Sorting; Search trees; Heaps; Hashing; Greedy approach; Dynamic programming; Graph algorithms; Shortest paths; Network flow; Disjoint Sets; Polynomial and matrix calculations; String matching; NP complete problems; Approximation algorithms.

Reference Material:

1. *Introduction to Algorithms /2E*, T. H. Cormen, C. E. Leiserson, and R. L. Rivest, MIT Press, McGraw-Hill, New York, NY, 2001.
2. Algorithms in C++; Robert Sedgewick

Course Name: Artificial Intelligence

Course Structure: Lectures: 2 / Labs: 3 **Credit Hours:** 3

Prerequisites: Discrete Structures

Objectives: This course studies four main objectives of AI. Modelling the environment by constructing computer representations of the real world. Perception and reasoning - obtaining and creating information/*knowledge* to populate a computational representation. Taking actions by using the knowledge of the environment and desired goals to plan and execute actions. Learning from past experience.

Course Outline: Artificial Intelligence: Introduction, Intelligent Agents. Problem-solving: Solving Problems by Searching, Informed Search and Exploration, Constraint Satisfaction Problems, Adversarial Search. Knowledge and reasoning: Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation. Planning and Acting in the Real World. Uncertain knowledge and reasoning: Uncertainty, Probabilistic Reasoning, Probabilistic Reasoning over Time, Making Simple Decisions, Making Complex Decisions. Learning: Learning from Observations, Knowledge in Learning, Statistical Learning Methods, Reinforcement Learning. Communicating, perceiving, and acting: Communication, Probabilistic Language Processing, Perception and Robotics. Introduction to LISP/PROLOG and Expert Systems (ES) and Applications.

Reference Material:

1. Artificial Intelligence: Structures and Strategies for Complex Problem Solving: International Edition By George F. Luger, 6th edition: Pearson Education, 2008.
2. Artificial Intelligence: A Modern Approach, By Stuart Jonathan Russell, Peter Norvig, John F. Canny, 2nd Edition, Prentice Hall, 2003.

Course Name: Computer Architecture

Course Structure: Lectures: 3, Labs: 0 **Credit Hours: 3**

Prerequisites: Digital Logic and Design

Objectives: Get a deeper understanding of how computers work, working knowledge of various subsystems and the general principles that affect their performance, analyze the performance of systems and quantify the performance measurements, fundamentals of all technologies, and advanced architectural features that boost the performance of computers.

Course Outlines: Fundamentals of Computer Design including performance measurements & quantitative principles, principles of Instruction Set Design, Operands, addressing modes and encoding, pipelining of Processors: Issues and Hurdles, exception handling features, Instruction-Level Parallelism and Dynamic handling of Exceptions, Memory Hierarchy Design, Cache Design, Performance Issues and improvements, Main Memory Performance Issues, Storage Systems, Multiprocessors and Thread Level Parallelism. Case Studies.

Resources:

1. *Computer Architecture: A Quantitative Approach* by Hennessy & Patterson, Morgan & Kauffman Series (2006) Fourth Edition.
2. *Computer Organization & Design : The Hardware/Software Interface* By Patterson & Hennessy, Morgan & Kauffman Series (2008) Fourth Edition.

Course Name: Compiler Construction

Course Structure: Lectures: 2 / Labs: 3 **Credit Hours: 3**

Prerequisites: Theory of Automata and Formal Languages

Objectives: At the end of the course students should understand the overall structure of a compiler, and will know significant details of a number of important techniques commonly used. They will be aware of the way in which language features raise challenges for compiler builders.

Course Outline: Compiler techniques and methodology. Organization of compilers. Lexical and syntax analysis. Parsing techniques. Object code generation and optimization, detection and recovery from errors. Contrast between compilers and interpreters.

Reference Material:

1. Compilers: Principles, Techniques, and Tools By Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Contributor Jeffrey D. Ullman ,Addison-Wesley Pub. Co., 2nd edition,1987 Original from the University of Michigan
2. Modern Compiler Design, By Dick Grune, Henri E. Bal, Criel J. H. Jacobs, Koen G. Langendoen, John Wiley, 2000.
3. Modern Compiler Implementation in C, By Andrew W. Appel, Maia Ginsburg, Contributor Maia Ginsburg, Cambridge University Press, 2004.
4. Modern Compiler Design by Dick Grune, Henri E. Bal, Criel J. H. Jacobs, Koen G. Langendoen, 2003, John Wiley & Sons.

Computer Science – Supporting Courses (9 credit hours)

Course Name: Multivariable Calculus

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: Calculus and Analytical Geometry

Objectives: The goals are to develop the skills to have ground knowledge of multivariate calculus and appreciation for their further computer science courses.

Course Outline: Functions of Several Variables and Partial Differentiation. Multiple Integrals, Line and Surface Integrals. Green's and Stoke's Theorem. Fourier Series: periodic functions, Functions of any period P-2L, Even & odd functions, Half Range expansions, Fourier Transform. Laplace Transform, Z-Transform.

Reference Material:

1. James Stewart, Multivariable Calculus, 6th edition, 2007, Cengage Learning publishers.
2. Swokowski, Olinick and Pence, *Calculus and Analytical Geometry*, 6th edition, 1994, Thomson Learning EMEA, Ltd.
3. Bernard Kolman, William F. Trench, Elementary Multivariable Calculus, 1971, Academic Press.
4. Howard Anton, Albert Herr, Multivariable Calculus, 5th edition, 1995, John Wiley.

Course Name: Differential Equations

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: Calculus and Analytical Geometry

Objectives: Develop fundamental skills of solving ordinary differential equations, and developing differential equations for real-world problems.

Course Outline: Ordinary Differential Equations of the First Order: Geometrical Considerations, Isoclines, Separable Equations, Equations Reducible to

Separable Form, Exact Differential Equations, Integrating Factors, Linear First-Order Differential Equations, Variation of Parameters. Ordinary Linear Differential Equations; Homogeneous Linear Equations of the Second Order, Homogeneous Second-Order Equations with Constant Coefficients, General Solution, Real Roots, Complex Roots, Double Root of the Characteristic Equation, Differential Operators, Cauchy Equation, Homogeneous Linear Equations of Arbitrary Order, Homogeneous Linear Equations of Arbitrary Order with Constant Coefficients, Non-homogeneous Linear Equations. Modelling of Electrical Circuits. Systems of Differential Equations. Series Solutions of Differential Equations. Partial Differential Equations: Method of Separation of variables, wave, Heat & Laplace equations and their solutions by Fourier series method.

Reference Material:

1. Michael Greenberg, *Advanced Engineering Mathematics*, 1996, Prentice Hall publishers.
2. Erwin Kreyzig, *Advanced Engineering Mathematics*, 7th edition, 1993, John Wiley & Sons Inc.
3. Zill, Prindle, Weber and Schmidt, *A First Course in Differential Equations*, 1996, Brooks/Cole Publishing,
4. Dennis G. Zill, Michael R. Cullen. *Differential Equations with Boundary-Value Problems*, 1996, Brooks/Cole Publishing,
5. C. H .Edwards, David E. Penney, *Elementary Differential Equations With Applications*, 1993, Prentice Hall.

Course Name: Numerical Computing

Course Structure: Lectures: 2 / Labs: 3 **Credit Hours:** 3

Prerequisites: Calculus and Analytical Geometry

Objectives: On completion of this unit, students will be able to demonstrate programming proficiency using structured programming techniques to implement numerical methods for solutions using computer-based programming techniques .using Matlab for all methods. The course must serve the purpose of scientific software development for science and engineering problems.

Course Outline: The concepts of efficiency, reliability and accuracy of a method. Minimising computational errors. Theory of Differences, Difference Operators, Difference Tables, Forward Differences, Backward Differences and Central Differences. Mathematical Preliminaries, Solution of Equations in one variable, Interpolation and Polynomial Approximation, Numerical Differentiation and Numerical Integration, Initial Value Problems for Ordinary Differential Equations, Direct Methods for Solving Linear Systems, Iterative Techniques in Matrix Algebra, Solution of non-linear equations.

Reference Material:

1. Numerical Methods in Scientific Computing Germund Dahlquist and Åke Björck .
2. Numerical Methods for Scientific Computing : J.H. Heinbockel
3. Numerical Analysis: I.A. Khubaza
4. Numerical Analysis and Programming : Shan S Kuo
5. Numerical Analysis by Berden Fairs

Computer Science – Elective Courses (21 credit hours)

Course Name: **Computer Graphics**

Course Structure: Lectures: 2 / Labs: 3 **Credit Hours:** 3

Prerequisites: Object Oriented Programming , Visual Programming

Objectives: Study of various algorithms in computer graphics and their implementation in any programming language.

Course Outline: Graphics hardware. Fundamental algorithms. Applications of graphics. Interactive graphics programming — graph plotting, windows and clipping, and segmentation. Programming raster display systems, Differential Line Algorithm, panning and zooming. Raster algorithms and software — Scan-Converting lines, characters and circles. Scaling, Rotation, Translation, Region filling and clipping. Two and three dimensional imaging geometry (Perspective projection and Orthogonal projection) and transformations. Curve and surface design, rendering, shading, colour and animation.

Reference Material:

1. Computer Graphics, Principles and Practice, J. D. Foley, A. van Dam, S. K. Feiner and J. F. Hughes, Addison-Wesley ISBN: 0-201-12110-7.
2. *Computer Graphics*, F.S.Hill, Maxwell MacMillan ISBN: 0-02-354860-6.
3. Interactive Computer Graphics: Functional, Procedural and Device-level methods; Peter Burger and Duncan. F. Gillies; Addison-Wesley, (2003)

Course Name: **Digital Image Processing**

Course Structure: Lectures:3 Labs: 0 **Credit Hours:** 3

Prerequisites:

Objective: The aim of this module is to understand the main terms & concepts of Information Systems & their applications in everyday business. The main objectives of this module are to make business students aware of the increasing importance of IT, computers and telecom and to manage IT systems in modern organisational structure. Another important objective of this module is to learn about various information systems used in industries and select the appropriate information system for the required application. Restoration in the Presence of Noise Only–Spatial Filtering, Mean Filters, Order-Statistics Filters, Adaptive Filters, Periodic Noise Reduction by Frequency Domain Filtering, Bandreject Filters, Bandpass Filters, Notch Filters. Estimating the Degradation Function, Estimation by Image Observation, Estimation by Experimentation, Estimation by Modeling, Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering. Image Segmentation, Detection of Discontinuities, Point Detection, Line Detection, Edge Detection, Edge Linking and Boundary Detection, Local Processing, Global Processing via the Hough Transform.

Thresholding, The Role of Illumination, Basic Global Thresholding, Basic Adaptive Thresholding, Local Thresholding, Thresholds Based on Several Variables.

Region-Based Segmentation, Region Growing, Region Splitting and Merging

Course Name: Digital Signal Processing

Course Structure: Lectures:3 Labs: 0 **Credit Hours:** 3

Prerequisites:

Objective: Introduction to signal, Analog and digital/discrete signal, Applications of DSP. Energy of a signal, Transformation Of independent variable, Signal types. Mathematical representation of periodic signal, Relation of complex no. with circle, Euler's relation, Difference b/w continuous time and discrete time signal. Unit impulse and unit step and their relationship, Convolution, Impulse response of an LTI system, Examples of an LTI system, Properties of an LTI system, Commulative property, Distributive property, Associative property, LTI system with and without memory, Invertability, Properties of an LTI system, Causality, Stability, Step response of an LTI system, Fourier series representation of periodic signal, Session 1 of Matlab, Session 2 of Matlab, Fourier series representation of periodic signals, Fourier series representation, Examples of Fourier series representation, Examples of Fourier series representation.

Reference Material:

1. Oppenheim, Signals and systems.
2. Oppen Heim, Discrete time signal Processing

Course Name: Computer Vision

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Pre-Req: Data Structures and Algorithms

Objectives: By the end of this course Students will be able to explain the concepts behind computer based recognition and the extraction of features from raster images. Students will also be able to illustrate some successful applications of vision systems and will be able to identify the vision systems limitations.

Course Outlines: Concepts behind computer-based recognition and extraction of features from raster images. applications of vision systems and their limitations. Overview of early, intermediate and high level vision, Segmentation: region splitting and merging; quadtree structures for segmentation; mean and variance pyramids; computing the first and second derivatives of images using the isotropic, Sobel and Laplacian operators; grouping edge points into straight lines by means of the Hough transform; limitations of the Hough transform; parameterisation of conic sections. Perceptual grouping: failure of the Hough transform; perceptual criteria; improved Hough transform with perceptual features; grouping line segments into curves. Overview of mammalian vision: experimental results of Hubel and Weisel; analogy to edge point detection and Hough transform; Relaxation labelling of images: detection of image features; Grouping of contours and straight lines into higher order features such as vertices and facets. Depth measurement in images.

1. Text Books/Reference Books:

2. Computer Vision: A Modern Approach, By David Forsyth, Jean Ponce, Prentice Hall, 2003.
3. Computer Vision, By Linda G. Shapiro, George C. Stockman, Prentice Hall, 2001.

4. Handbook of Mathematical Models in Computer Vision, By Nikos Paragios, Yunmei Chen, Olivier Faugeras, Birkhäuser, 2006.

Course Name: Software Engineering

Course Structure: Lectures: 3hrs **Credit Hours:** 3

Objective: The students will study techniques for software verification, validation and testing. They would also study reliability and performance issues in software design and development. Upon successful completion of this course the student will be to understand the importance of software engineering to computer science and the most important general approaches to structuring the software production process, analyze the requirements for a software system and produce a software design from requirements (Data Flow Diagram (DFD)), assess software productivity using metrics, use different testing techniques used in software engineering to test software systems, manage the important issues for planning a project.

Course Outlines:

- Introduction to Software Engineering,
- Software Process Framework
- Process Models
- Agile Software Process
- Software Engineering Practices
- System Engineering
- Requirement Engineering
- Analysis Modelling
- Design Engineering
- Architectural Design
- Component Design
- User Interface Design
- Testing Strategies
- Testing Tactics
- Product and Process Metrics
- Project Management
- Project Estimation
- Project Scheduling
- Risk Management
- Quality Management
- Change Management

Text Books/Reference Books:

1. *Software Engineering: A Practioner's Approach*, Roger Pressman, McGraw-Hill, Sixth Edition, 2005.
2. Ian Sommerville. *Software Engineering*, Addison-Wesley, 2001 (7th edition).
3. UML Distilled

Course Name: Data Communication

Course Structure: 3 **Credit Hours:** 3

Pre-requisite:

Objectives: To provide knowledge of Data Communication and different

mechanisms of communication

Course Outlines: Introduction, Data and Network, Layers, OSI Model, Introduction to Signals, Transmission Media, Digital Transmission, PAM, PCM, ASK, FSK, PSK, QAM, Data Communication Techniques and technologies, Modulation, Multiplexing, Types of errors, Data Communication Protocols, Current technologies being used for data communication.

Reference Material:

1. Behrouz A. Forouzan, Data Communication and Networking, 3rd Edition.
2. William Stalling, Business Data Communication.

Course Name: Distributed Computing

Course Structure: Lectures: 3 Labs: 0 **Credit Hours: 3**

Course Outlines: Why use parallel and distributed systems? Why not use them? Speedup and Amdahl's Law, Hardware architectures: multiprocessors (shared memory), networks of workstations (distributed memory), clusters (latest variation). Software architectures: threads and shared memory, processes and message passing, distributed shared memory (DSM), distributed shared data (DSD). Possible research and project topics, Parallel Algorithms, Concurrency and synchronization, Data and work partitioning, Common parallelization strategies, Granularity, Load balancing, Examples: parallel search, parallel sorting, etc. Shared-Memory Programming: Threads, Pthreads, Locks and semaphores, Distributed-Memory Programming: Message Passing, MPI, PVM. Other Parallel Programming Systems, Distributed shared memory, Aurora: Scoped behaviour and abstract data types, Enterprise: Process templates. Research Topics

Text Books/Reference Books:

1. B. Wilkinson and M. Allen, *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*, 1/e, Prentice Hall, 1999.
2. W. Stevens, *Advanced Programming in the Unix Environment*, Addison Wesley, 1993.

Course Name: Data and Network Security

Course Structure: Lectures: 3 Labs: 0 **Credit Hours: 3**

Course Outlines: Introduction; Cryptology and simple cryptosystems; Conventional encryption techniques; Stream and block ciphers; DES; More on Block Ciphers; The Advanced Encryption Standard. Confidentiality & Message authentication: Hash functions; Number theory and algorithm complexity; Public key Encryption. RSA and Discrete Logarithms; Elliptic curves; Digital signatures. Key management schemes; Identification schemes; Dial-up security. E-mail security, PGP, S-MIME; Kerberos and directory authentication. Emerging Internet security standards; SET; SSL and IPsec; VPNs; Firewalls; Viruses; Miscellaneous topics.

Text Books/Reference Books:

1. W. Stallings, *Cryptography and Network Security*, Prentice Hall PTR, Upper Saddle River, NJ, 2003.

2. Kaufman, R. Perlman, M. Speciner, *Network Security: Private Communication in a Public World* – Prentice Hall PTR, Upper Saddle River, NJ, 2002.
3. M. Bishop, *Computer Security: Art and Science* – Addison-Wesley, 2003.
4. Stinson, *Cryptography: Theory and Practice*, CRC Press, Boca Raton, FL, 1995.
5. Richard A. Mollin, *An Introduction to Cryptography*, Chapman and Hall/CRC, 2001.
6. B. Schneier, *Applied Cryptography*, John Wiley and Sons, NY, 1996.
7. A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.

Course Name: Wireless Networks

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outlines: This course covers fundamental techniques in design and operation of first, second, and third generation wireless networks: cellular systems, medium access techniques, radio propagation models, error control techniques, handoff, power control, common air protocols (AMPS, IS-95, IS-136, GSM, GPRS, EDGE, WCDMA, cdma2000, etc), radio resource and network management. As an example for the third generation air interfaces, WCDMA is discussed in detail since it is expected to have a large impact on future wireless networks. This course is intended for graduate students who have some background on computer networks.

Text Books/Reference Books:

1. Theodore S Rappaport, *Wireless Communications*.
2. David Tse, *Fundamentals of Wireless Communications*.
3. W. Stallings, *“Wireless Communications and Networks”*, Prentice Hall, 2002.
4. T.S. Rappaport, *“Wireless Communications: Principles & Practice”*, Second Edition, Prentice Hall, 2002.
5. J. Schiller, *“Mobile Communications”*, Addison Wesley, 2000.
6. V.K. Garg, *“IS-95 CDMA and cdma 2000”*, Prentice Hall PTR, 2000.
7. J.P. Castro, *“The UMTS Network and Radio Access Technology - Air Interface Techniques for Future Mobile Systems”*, Wiley, 2001.
8. H. Holma and A. Toskala, *“WCDMA for UMTS Radio Access for Third Generation Mobile Communications”*, John Wiley & Sons, 2001.

Course Name: Telecommunication Systems

Course Structure: Lectures:3 Labs: 0/3 **Credit Hours:** ¾

Prerequisites: None

Objectives: To provide a first level exposure to the broad domain of telecommunication Systems

Course Outline: Introduction to media, bandwidth and noise. Twisted pair (UTP, STP), coaxial cables (types and specifications), optical fibres (types and losses), Introduction to optical sources and detectors. Microwave links, satellite communication and infrared links. Frequency Division Multiplexing (FDM), TDM, FDMA, TDMA and CDMA. Switching: circuit and packet switching. Introduction to mobile and cellular communications. Block diagram and current trends.

Reference Material:

1. *Introduction to Telecommunications Network Engineering*, 2nd edition, T. Aattalainen, Artech House 2003, ISBN: 1580535003.
2. *Fundamentals of Telecommunication Networks*, T. Saadawi, Wiley US, ISBN: 0471515825.
3. *Telecommunication Systems*, P. G. Fonteolliet, Artech House 1991.

Course Name: System Programming

Course Structure: Lectures: 2 / Labs: 3 **Credit Hours: 3**

Prerequisites: Operating Systems

Objectives: Demonstrate mastery of the internal operation of Unix system software including assemblers, loaders, macro-processors, interpreters, inter-process communication.

Course Outline: System Programming overview: Application Vs. System Programming, System Software, Operating System, Device Drivers, OS Calls. Window System Programming for Intel386 Architecture: 16 bit Vs 32 bit, Programming, 32 bit Flat memory model, Windows Architecture. Virtual Machine (VM) Basics, System Virtual Machine, Portable Executable Format, Ring 0 Computer, Linear Executable format, Virtual Device Driver (V + D), New Executable format, Module Management, COFF obj format 16 bit. (Unix) other 32-bit O.S Programming for I 386; Unix Binary format (ELF), Dynamic shared objects, Unix Kernel Programming (Ring 0), Unix Device Architecture (Character & Block Devices), Device Driver Development, Enhancing Unix Kernel.

Reference Material:

1. *The UNIX Programming Environment*, B. Kernighan & R. Pike Prentice-Hall, 1984.
2. *System Software*, Leland L. Beck, Addison-Wesley Longman, 1990, ISBN: 0-201-50945-8.

Course Name: Distributed Database System

Course Structure: Lectures: 3 (3,0) **Credit Hours: 3**

Prerequisites: Database Systems

Objectives: To clearly describe the difference of Centralized database and Distributed database and enable the students to design/model a distributed database.

Course Outline: Introduction, Overview of relational DBMS and Normalization, Distributed DBMS architecture, Distributed database design and Data Distribution Strategies, Replication/Fragmentation, Distributed Transaction Management, Distributed Query Processing, Distributed Concurrency Control, Distributed Data Security, Distributed Database Recovery.

Reference Material:

1. *Principals of Distributed Database Systems* by Ozsu Tamer.
2. *Database Systems* by Thomas Connolly.

Course Name: Datawarehouse

Course Structure:

Credit Hours: 3

Prerequisite: Database Systems

Objective: To provide the Introduction of Datawarehouse and its purpose. And enable the students to understand different features / issues in datawarehousing and its designing.

Course Outline: Introduction to Data Warehouse and Data Marts, Comparison of OLTP Systems & Data Warehousing, Data Warehouse Architecture, Dimensional Modeling, Comparison Of DM & ER Models, Extraction, Cleansing and Loading process and techniques, Designing a Data warehouse, End user tools, OLAP.

Course Name: Entrepreneurship

Course Structure: Lectures: 3 Labs: 0

Credit Hours: 3

Pre-requisite: Introduction to Management, **Semester: 8**
Principles of Accounting

Course Outlines: This course provides the student with an understanding of the entrepreneurship process. It exposes them to the concepts, practices and tools of the entrepreneurial world. This will be accomplished through a combination of readings, cases studies and projects designed to convey the unique environment of the entrepreneurs and new ventures. The course gives students the tools necessary to think creatively, to plan out whether their idea is marketable to investors, guide them through the launch their own business, or to support an employer in launching and growing an entrepreneurial venture. As CS students, the focus shall be on items particularly important for technology ventures.

Text Books/Reference Books:

1. The Art of the Start: The Time-Tested, Battle-Hardened Guide for Anyone Starting Anything by Guy Kawasaki, ISBN: 1591840562.

MS (Computer Science)

Curriculum for MS (Computer Science)

The recommendations of the last meeting held in August 2008 were also considered and very minor changes in the structure have been made in the light of committee's recommendations. The complete detail regarding proposed MS (Computer Sciences) Programme is available herein the following pages

- Minimum credit hours shall be 30 for MS (Computer Science) programme.
- The programme shall comprise 4 semesters spread over 2 years with two semesters a year.
- The additional major areas have been appended in the list of specialization each having on average 4 courses from “**Computer Science Curriculum 2008: Interim Revision of CS2001 report**” by Interim Review Task Force of ACM and IEEE Computer Society.

The following is the modified distribution of total credit hours:

Category or Area	Credit Hours
Core	12
Electives	12
Thesis	9
Total Credit Hours	33

Objectives

A challenging graduate programme may be structured on the basis of the classical objective, which is the preparation for study of doctoral level, and this remains an important aspect of such programmes, but it is believed that all programmes should prepare the student for study beyond the master's level.

Many people already in the field desire additional training in Computer Science. These individuals may have undergraduate degrees in Computer Science and desire to advance; or they may have considerable experience in Computer Science, but little formal education in the field. While this latter group should be declining in number as more undergraduate Computer Science majors enter the job market, the demand does exist and will continue to do so in the foreseeable future. In addition, there will be a continuing need for individuals with a bachelor's degree in Computer Science to update their training.

Among the objectives for students in master's programmes is entry into the Computer Science field at a relatively high level of responsibility and expertise. Computer Science is such a new and rapidly expanding field that individuals entering with a master's degree in this field will almost immediately move to positions with great responsibility. This, in turn, implies the requirement for an advanced level of prior training in both technical and related areas (e.g. communication skills). In all these cases, the master's degree provides both motivations for the student and a standard for reward by the employer.

Programme Structure

The graduate programme should embody sufficient flexibility to fulfil the requirements of either an "academic" degree (Breadth-Based) obtained in preparation for further graduate study or a terminal "professional" degree (Depth-Based). The discipline of Computer Science has matured enough that the distinction between academic and professional programmes is beginning to appear. However, the concept of an utterly terminal programme is not widely accepted in the field. All Computer Science academic programmes should provide the possibility of additional study in the field. The proposed programme is intended to establish an integrated breadth and depth based curriculum model to assure that the common aspects of various potential masters' programmes in Computer Science are captured.

The proposed curriculum structure may be implemented within four-semester time. A project/thesis work may be unified with student's chosen depth oriented specialties. Generally graduate programmes are structured with a common core of fundamental material and wide range of options for the rest of the course work.

Eligibility

- ***BS (CS) 4 Years Degree Programme (min 130 credit hours), or***
- **Computer Science Conversion Course 2 Years Degree Programme referred to as "MCS" or "MSc (CS)".**
- **BCS-3 years Degree Programme-Student will be required to complete the deficiency of difference of total earned credit hours and 130 credit hours.**
- **16 year Science and Engineering graduates are eligible but they have to cover deficiency.**

SCHEME OF STUDIES MS (CS)

Courses Requirements:

Core courses

#	Code	Course Title	Credit hours	Semester
1	CS	Advanced Theory of Computation	3	1
2	CS	Advanced Algorithm Analysis	3	1
3	CS	Advanced Operating Systems	3	1
4	CS	Advanced Computer Architecture	3	1
(12/30)				

Electives (Specialized Areas)-Not limited to the list given below (4 Courses of 12 credit hours)

Code	Specialization Areas	Crt. Hrs	Code	Specialization Areas	Crt.Hrs
	Software Engineering			Artificial Intelligence	
CS	Advanced Software Development	3	CS	Design of Intelligent Systems	3
CS	Topics in Software Engineering	3	CS	Machine Learning	3
CS	Object Oriented Software Engineering.	3	CS	Neural Networks	3
CS	Software Quality Assurance	3	CS	Mathematical Reasoning	3
CS	Requirements Engineering	3	CS	Decision Support Systems	3
CS	Software Architecture	3	CS	Computer Vision	3
CS	Agent Oriented Software Engineering	3	CS	Automated Reasoning	3
CS	Software Project Management	3	CS	Knowledge based systems	3
CS	Software Design	3	CS	Planning systems	3
CS	Software Engineering and Formal Specifications	3	CS	Natural Language Processing	3
CS	Empirical Software Engineering	3	CS	Agents	3
CS	Software Process Improvement	3	CS	Robotics	3
CS	Component-Based Computing	3	CS	Symbolic Computation	3
CS	Programming	3	CS	Genetic Algorithms	3

	Environment				
CS	Safety-Critical Systems	3		CS	Semantic Web 3
	Information Management				Computer Architecture and Organization
CS	Advanced DBMS	3		CS	Embedded Systems 3
CS	Multimedia Information Systems	3		CS	Parallel and Distributed Systems 3
CS	Database Design	3		CS	Design Verification 3
CS	Transaction Processing	3		CS	Integrated Circuit 3
CS	Distributed and Object Databases	3		CS	System on a chip 3
CS	Data Mining	3		CS	VLSI Development 3
CS	Spatial and Temporal Databases	3		CS	Device Development 3
CS	Semantic Databases	3			
CS	Data Warehousing	3			Graphics and Visual Computing
CS	Object Oriented Databases	3		CS	Advanced Computer Graphics 3
CS	Digital Libraries	3		CS	Multimedia & Hypermedia System 3
CS	Web-Based DBMS	3		CS	Virtual Reality 3
CS	Topics in DBMS	3		CS	Visualization 3
CS	Data Grids	3		CS	Geographical Information Systems 3
CS	Text Mining	3		CS	Computer Animation 3
				CS	Genetic Algorithms 3
	System Engineering			CS	Human Computer Interaction 3
CS	Digital Signal Processing	3			
CS	Switching and Fault Diagnosis	3			Computer Science Education
CS	FPGAs and Verilog	3		CS	Educational Technology 3
CS	Control Systems and Robotics	3		CS	Multimedia and Hypermedia Sys 3
CS	Real Time Systems	3		CS	Computer Aided Instructions 3
CS	Parallel & Distributed Systems	3		CS	Web Based Education Systems 3
CS	Control Systems and Robotics	3		CS	Measurement of Learning 3
CS	Real Time Operating Systems	3		CS	Topics in Comp Science Education 3
CS	Embedded System	3			

CS	ASIC Design	3			
CS	VHDL	3			Human Computer Interaction
				CS	Intelligent User Interfaces 3
	Net-Centric computing			CS	Information Retrieval Techniques 3
CS	Advanced Computer Networks	3		CS	Rich Internet Applications 3
CS	Network Security	3		CS	Graphical User Interfaces 3
CS	Topics in Computer Networking	3		CS	Computer-Supported Cooperative Work (CSCW) 3
CS	Broadband and Satellite Communication	3		CS	Multimedia Systems Development 3
CS	Mobile and Pervasive computing	3		CS	Interactive-Systems Development 3
CS	Wireless and Mobile Computing Networks	3			
CS	Intelligent and active networks	3			Social and Professional Issues (SP)
CS	Network Performance Evaluation	3		CS	Social Context computing 3
CS	Cluster Computing	3		CS	Computing and Ethics 3
CS	Distributed Computing	3		CS	Computing Economics 3
CS	Data Compression	3		CS	Computer Law 3
CS	Network Management	3		CS	Intellectual Property 3
CS	Enterprise Networking	3		CS	Privacy and Civil Liberties 3
CS	Programming for the World-Wide Web	3			
					Operating Systems
	Programming Language Design and Translators			CS	Concurrent and Distributed Systems 3
CS	Compiler Construction			CS	Dependent Computing 3
CS	Programming Language Design	3		CS	Fault- Tolerance 3

CS	Programming Language Semantics	3		CS	Real- Time Systems	3
CS	Programming Paradigms	3				
CS	Functional Programming	3			Discrete Structures	
CS	Logic Programming	3		CS	Combinatorics	3
CS	Scripting Languages	3		CS	Probability and Statistics	3
CS	Algorithm and complexity (AL)	3		CS	Coding and Information Theory	3
CS	Advanced Algorithmic Analysis	3				
CS	Automata and Language Theory	3			Computational Science	
CS	Cryptography	3		CS	Computational Science	3
CS	Geometric Algorithms	3		CS	Numerical Analysis	3
CS	Parallel Algorithms	3		CS	Operations Research	3
				CS	Simulation and Modelling	3
	Grid and Cloud Computing			CS	Scientific Computing	3
CS	Autonomous Computing	3		CS	Computational Biology	3
CS	Data Grids	3				
CS	Semantic Grid	3			Web Engineering	
CS	Computational Grid	3		CS	Semantic Web	3
CS	Utility Computing	3		CS	Web Services	3
CS	Autonomous Computing	3				
CS	Data Grids	3				

Model Programme: Semester-wise Plan MS (CS)

Semester 1 (12 credit hrs)

Subjects			Credit Hrs
1	CS	Advanced Theory of Computation	3
2	CS	Advance Algorithm Analysis	3
3	CS	Advanced Operating Systems	3
4	CS	Advanced Computer Architecture	3
			Total: 12

Semester 2 (9 credit hrs)

Subjects			Credit Hrs
1	CS	Elective I	3
2	CS	Elective II	3
3	CS	Elective III	3
			Total: 9

Semester 3 (4 credit hrs)

Subjects			Credit Hrs
1	CS	Thesis (partial registration)	3
2	CS	Elective IV	3
			Total: 6

Semester 4 (5 credit hrs)

Subjects			Credit Hrs
1	CS	Thesis (partial registration)	6
			Total: 6

Total (all semesters) = 33

Course Description and Profiles:

Core Courses:

Course Name: Advanced Theory of Computation

Course Structure: Lectures: 3 Labs: 0 **Credit Hours: 3**

Course Outline: Automata theory, formal languages, Turing machines, computability theory and reducibility, computational complexity, determinism, non-determinism, time hierarchy, space hierarchy, NP completeness, selected advanced topics.

Text Books/Reference Books:

1. Michael Sipser, *Introduction to the Theory of Computation*, First Edition, 1997, PWS Publishing Company.
2. Christos Papadimitriou, *Computational Complexity*, 1994, Addison-Wesley.
3. John Hopcroft and Jeffrey Ullman, *Introduction to Automata Theory, Languages, and Computation*, 1979, Addison-Wesley. (or the second edition).
4. Tao Jiang, Ming Li, and Bala Ravikumar, Formal models and Computability, in *Handbook of Computer Science*, CRC Press, 1996.
5. T.H. Cormen, et al., *Introduction to Algorithms*, MIT Press and McGraw-Hill Book Co., 1990.
6. Peter Linz, *An Introduction to Formal Languages and Automata*, ISBN: 0-669-17342-8.

Course Name: Advance Algorithm Analysis

Course Structure: Lectures: 3 Labs: 0 **Credit Hours: 3**

Pre-Req: Data Structures and Algorithms

Course Outline: Advanced algorithm analysis including the introduction of formal techniques and the underlying mathematical theory. NP-completeness. Search Techniques. Randomized Algorithms. Heuristic and Approximation Algorithms. Topics include asymptotic analysis of upper and average complexity bounds using big-O, little-o, and theta notation. Fundamental algorithmic strategies (brute-force, greedy, divide-and-conquer, backtracking, branch-and-bound, pattern matching, and numerical approximations) are covered. Also included are standard graph and tree algorithms. Additional topics include standard complexity classes, time and space tradeoffs in algorithms, using recurrence relations to analyze recursive algorithms, non-computable functions, the halting problem, and the implications of non-computability. Algorithmic animation is used to reinforce theoretical results. Upon completion of the course, students should be able to explain the mathematical concepts used in describing the complexity of an algorithm, and select and apply algorithms appropriate to a particular situation.

Text Books/Reference Books:

1. Approximation Algorithms, By Vijay V. Vazirani, Springer, 2004.
2. Introduction to Algorithms, By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, 2nd edition, Published by MIT Press, 2001.
3. Algorithms and Theory of Computation Handbook, By Mikhail J. Atallah Contributor Mikhail J. Atallah, CRC Press, 1998.

Course Name: Advance Operating System

Course Structure: Lectures:3 Labs: 0 **Credit Hours:** 3

Prerequisites:

Objective: To apprise the students with characteristics of modern operating systems and architectural models.

Course Contents Course Outline:

- Introduction
 - Characterization of Modern Operating Systems; file systems, memory management techniques, Process scheduling and resource management,
- System Models
 - Architectural models
- Interprocess Communication
- Issues of Security in Distributed Systems (Partial coverage)
- Distributed File System
- Concurrency Control in Distributed Systems
- Problems of coordination and agreement in Distributed Systems
- Replication – Advantages and requirements
 - Fault-tolerant services
- Mobile and Ubiquitous Computing

Reference Books

1. Distributed Systems Concepts and Design 4th edition by George Coulouris, Jean Dollimore and Ttim Kindberg
2. Distributed Operating Systems: Concepts and Design by Pradeep k. Sinha
3. Advanced Concepts in Operating Systems by Singhal and Shiviratri

Course Name: Advance Computer Architecture

Course Structure: Lectures:3 Labs: 0 **Credit Hours:** 3

Prerequisites: Computer Architecture

Objective: To develop a thorough understanding of high-performance computer architecture, as a foundation for advanced work in computer architecture.

Course Outlines: This course is aimed at the hardware aspects of parallel computer architectures including the design and protocols evaluation for memory coherence, inter-connection networks and system scalability. Advanced topics in this course will cover multiprocessors on a chip, reconfigurable computing and power aware designs. Various coarse-grained and fine-grained architectures with reference to SIMD and MIMD designs should also be covered.

Text book:

1. “Advanced Computer Architecture: A Design Space Approach”, Dezso Sima, Terence Fountain, Peter Kacsuk, Addison-Wesley Publishers, 1997.
2. “Scalable Parallel Computing Technology, Architecture, Programming”, Kai Hwang, Zhiwei Xu, McGraw Hill Publishers, 1998.

Electives Courses

Course Name: Digital Signal Processing

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: One- and N -dimensional signals and systems, Sampling theorem, Discrete-time Fourier transform, discrete Fourier transform, fast Fourier transform, z-transforms: stability and minimum phase signals/systems, *Linear filtering of signal*: Time domain: Difference equations and convolution, Impulse invariance, bilinear transform, FIR filter design, 2D filter design, *Statistical signal processing*: Stochastic signals: correlation functions and power density spectra, Optimal filtering: Wiener filters, Adaptive filters: LMS and array processing.

Text Books/Reference Books:

1. *Discrete-Time Signal Processing*, 2nd edition Alan V. Oppenheim and Ronald W. Schaffer, Prentice-Hall.

Course Name: Parallel and Distributed Computing

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outlines: Why use parallel and distributed systems? Why not use them? Speedup and Amdahl's Law, Hardware architectures: multiprocessors (shared memory), networks of workstations (distributed memory), clusters (latest variation). Software architectures: threads and shared memory, processes and message passing, distributed shared memory (DSM), distributed shared data (DSD). Possible research and project topics, Parallel Algorithms, Concurrency and synchronization, Data and work partitioning, Common parallelization strategies, Granularity, Load balancing, Examples: parallel search, parallel sorting, etc. Shared-Memory Programming: Threads, Pthreads, Locks and semaphores, Distributed-Memory Programming: Message Passing, MPI, PVM. Other Parallel Programming Systems, Distributed shared memory, Aurora: Scoped behaviour and abstract data types, Enterprise: Process templates. Research Topics.

Text Books/Reference Books:

1. B. Wilkinson and M. Allen, *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*, 1/e, Prentice Hall, 1999.
2. W. Stevens, *Advanced Programming in the Unix Environment*, Addison Wesley, 1993.

Course Name: Control Systems and Robotics

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: Review of classical control analysis methods. Nyquist stability criterion. Classical design using frequency domain methods, phase lead and lag controllers, PID controllers. Relay auto tuning. Introduction to state space methods. State space models, state transformations, solution of the state equations. Controllability and observability. Design using state feedback. LQR design, pole placement, use of observers. Introduction to robotics. Transducers, actuators and robot control.

Text Books/Reference Books:

1. R.C. Dorf, Modern Control Systems, 7th (1995), 8th (1998) or 9th (2001) Edition, Addison-Wesley.
2. C.C. Bissell, Control Engineering, 2nd Edition, 1994, Publisher: Chapman & Hall.
3. K.Ogata, *Modern Control Engineering*, Prentice Hall, 2nd ed. 1990.

Course Name: Real Time Operating Systems

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: The principles of real-time and embedded systems inherent in many hardware platforms and applications being developed for engineering and science as well as for ubiquitous systems, including robotics and manufacturing, interactive and multimedia, immersive and omnipresent applications. Real-time and quality of service system principles, understand real-time operating systems and the resource management and quality of service issues that arise, and construct sample applications on representative platforms. Platforms range from handheld and mobile computers to media and real-time server systems. Platforms may also include specialized systems used in application-specific contexts, such as autonomous robotics, smart sensors, and others.

Text Books/Reference Books:

It is an advanced course and the instructor may make his notes from various resources at the web.

Course Name: Advanced Networking

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: Review of basic concepts: The OSI Model, packet and circuit switching, network topology, ISDN. The TCP/IP protocol stack: IP, ARP, TCP and UDP, DNS, ICMP, Internet Addressing, Routing, IP Multicast, RSVP, Next Generation IP – Ipng, Wireless: Radio basics, Satellite Systems, WAP, current trends, Issues with wireless over TCP. Congestion Control: Control vs. Avoidance. Algorithms, Congestion in the Internet. Mobile IP, Voice over IP (VoIP), VPNs, Network Security. Management: Quality of Service (QoS), network vs. distributed systems management Protocols, web-based management

Text Books/Reference Books:

1. James F. Kurose and Keith W. Ross, "Computer Networking – A Top-Down Approach Featuring the Internet", Addison Wesley.
2. Coulouris, Dollimore, Kindberg, "Distributed Systems – Concepts and Design", Addison Wesley.
3. William Stallings, "Data and Computer Communications", Prentice-Hall — Sixth Edition (for those who want to review basics of networking).

Course Name: Network Security

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: Introduction; Cryptology and simple cryptosystems; Conventional encryption techniques; Stream and block ciphers; DES; More on Block Ciphers; The Advanced Encryption Standard. Confidentiality & Message authentication: Hash functions; Number theory and algorithm complexity; Public key Encryption. RSA and Discrete Logarithms; Elliptic curves; Digital signatures. Key management schemes; Identification schemes; Dial-up security. E-mail security, PGP, S-MIME; Kerberos and directory authentication. Emerging Internet security standards; SET; SSL and IPsec; VPNs; Firewalls; Viruses; Miscellaneous topics.

Text Books/Reference Books:

1. W. Stallings, *Cryptography and Network Security*, Prentice Hall PTR, Upper Saddle River, NJ, 2003.
2. Kaufman, R. Perlman, M. Speciner, *Network Security: Private Communication in a Public World* – Prentice Hall PTR, Upper Saddle River, NJ, 2002.
3. M. Bishop, *Computer Security: Art and Science* – Addison-Wesley, 2003.
4. Stinson, *Cryptography: Theory and Practice*, CRC Press, Boca Raton, FL, 1995.
5. Richard A. Mollin, *An Introduction to Cryptography*, Chapman and Hall/CRC, 2001.
6. B. Schneier, *Applied Cryptography*, John Wiley and Sons, NY, 1996.
7. A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.

Course Name: Topics in Computer Networking

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: This course offers an advanced introduction and research perspectives in the areas of switch/router architectures, scheduling for best-effort and guaranteed services, QoS mechanisms and architectures, web protocols and applications, network interface design, optical networking, and network economics. The course also includes a research project in computer networking involving literature survey, critical analysis, and finally, an original and novel research contribution. Typical topics can be listed below:

Overview of packet switching networks and devices. Fundamentals of Internet Protocol (IP) networking. Route lookup algorithms. Router architecture and performance. Detailed operation of Internet routing protocols such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). Integrated and

differentiated network service models. Traffic Engineering (TE) concepts and mechanisms including label assignment, label distribution, and constraint-based routing algorithms. Multi-protocol label switching and its generalization. Quality of service mechanisms for multimedia and real-time communications. TE-based routing and signalling protocols. Fundamentals of per-flow and aggregate scheduling algorithms. Application-level and network-level signalling protocols for data, voice, and video communications. Resource signalling and resource reservation protocols. Worst-case analysis for multimedia networking.

Text Books/Reference Books:

1. Puzmanov, *Switching and Routing*, Addison Wesley, 2002.
2. Garica and Widjaja, *Communication Networks: Fundamentals Concepts and Key Architectures*, McGraw-Hill, 2001.
3. Peterson and Davie, *Computer Networking a Systems Approach*, 3rd Edition, Morgan Kaufman, 2003.
4. William Stallings, *High-Speed Networks: TCP/IP and ATM Design Principles*, Prentice Hall; 1998, ISBN: 0135259657.
5. Andrew S. Tanenbaum, *Computer Networks, 3rd Edition*. Prentice Hall, March 1996.

Course Name: Network Administration

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: Through completion of this course, students will be able to plan, install, and configure a Web Server, manage, monitor, and optimize a Web Server, and design and implement a Web Site on the Web Server created.

Text Books/Reference Books:

1. *Information Technology Project Management*. (2002) Course Technology. ISBN: 0-619-03528-5.
2. *Principles of Web Design*. (2000) Course Technology. ISBN: 0-619-01526-8.

Course Name: Wireless Networks

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: This course covers fundamental techniques in design and operation of first, second, and third generation wireless networks: cellular systems, medium access techniques, radio propagation models, error control techniques, handoff, power control, common air protocols (AMPS, IS-95, IS-136, GSM, GPRS, EDGE, WCDMA, cdma2000, etc), radio resource and network management. As an example for the third generation air interfaces, WCDMA is discussed in detail since it is expected to have a large impact on future wireless networks. This course is intended for graduate students who have some background on computer networks.

Text Books/Reference Books:

1. W. Stallings, "Wireless Communications and Networks", Prentice Hall, 2002.
2. T.S. Rappaport, "Wireless Communications: Principles & Practice", Second Edition, Prentice Hall, 2002.
3. J. Schiller, "Mobile Communications", Addison Wesley, 2000.
4. V.K. Garg, "IS-95 CDMA and cdma 2000", Prentice Hall PTR, 2000.

5. J.P. Castro, "The UMTS Network and Radio Access Technology - Air Interface Techniques for Future Mobile Systems", Wiley, 2001.
6. H. Holma and A. Toskala, "WCDMA for UMTS Radio Access for Third Generation Mobile Communications", John Wiley & Sons, 2001.

Course Name: Network Performance Evaluation

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: This is an advanced course in networks and protocols. Analytical, simulation and experimental methods should be used to evaluate and design networks and protocols. Investigate network management tools and techniques.

Text Books/Reference Books:

1. T. G. Robertazzi, Computer Networks and Systems: Queuing Theory and Performance Evaluation, Springer-Verlag, 2nd edition, 1994.

Course Name: Theory of Programming Languages

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: Introduction and History, Syntax and Semantics, Control Structures, Types, Logic Programming, Functional Programming and Lambda calculus, Concurrent and Distributed Programming, Dataflow, Object-oriented Programming.

Text Books/Reference Books:

1. Raphael Finkel, *Advanced Programming Language Design*, Addison-Wesley. ISBN: 0805311912
2. Introduction to the Theory of Programming Languages -- Bertrand Meyer
3. The Study of Programming Languages -- Ryan Stansifer
4. The Anatomy of Programming Languages -- Fischer and Grodzinsky
5. Concepts of Programming Languages -- Sebesta

Course Name: Advanced Compiler Design I

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: An in-depth study of compiler backend design for high-performance architectures. Topics include control-flow and data-flow analysis, classical optimization, instruction scheduling, and register allocation. Advanced topics include memory hierarchy management, optimization for instruction-level parallelism, modulo scheduling, predicated and speculative execution. The class focus is processor-specific compilation techniques, thus familiarity with both computer architecture and compilers is recommended.

Text Books/Reference Books:

1. *Compilers: Principles, Techniques, and Tools*, Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Addison-Wesley, 1988.
2. *Advanced Compiler Design & Implementation*, Steven S. Muchnick, Morgan Kaufmann, 1997.

3. *Building an Optimizing Compiler*, Robert Morgan, Butterworth-Heinemann, 1998.

Course Name: Advanced Compiler Design II

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: The course should consist of one or two major projects. Theoretical study should depend on the level of the first course Design I and the student needs.

Text Books:

1. *Compilers: Principles, Techniques, and Tools*, Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Addison-Wesley, 1988.
2. *Advanced Compiler Design & Implementation*, Steven S. Muchnick, Morgan Kaufmann, 1997.
3. *Building an Optimizing Compiler*, Robert Morgan, Butterworth-Heinemann, 1998.

Course Name: Intelligent User Interfaces

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: The increasing complexity of software and the proliferation of information makes intelligent user interfaces increasingly important. The promise of interfaces that are knowledgeable, sensitive to our needs, agile, and genuinely useful has motivated research across the world to advance the state of the art and practice in user interfaces that exhibit intelligence. The text covers the topic well.

Text Books:

1. *Readings in Intelligent User Interfaces*, Mark T. Maybury (Editor), Wolfgang Wahlster (Editor), Paperback - 736 pages (April 1998) Morgan Kaufman Publishers; ISBN: 1558604448.

Course Name: Multimedia Database

Course Structure: Lectures: 3 Labs: 0 **Credit Hours:** 3

Course Outline: Introduction; Overview of Relational and Object-Relational Data Representations; Text/Document Databases; Multidimensional Data Structures, similarity based search (spatial, image, audio); XML Databases; Temporal Data Models; Logical Frameworks.

Text Books/Reference Books:

1. *Principles of Multimedia Database Systems*, by V.S. Subrahmanian, Morgan Kaufmann Publishing Company, San Fransisco, CA. 1998. ISBN: 1558604669.
2. *Principles of Database Query Processing for Advanced Applications* (Morgan Kaufmann Series in Data Management Systems), by Clement T. Yu, Weiyi Meng, 1998. ISBN: 1558604340.
3. *Databases and Transaction Processing, An Application-Oriented Approach*, Philip M. Lewis, Arthur Bernstein, and Micheal Kifer. Addison Wesley

Course Name: Computer Vision

Course Structure: Lectures: 3 Labs: 0 **Credit Hours: 3**

Pre-Req: Data Structures and Algorithms

Objectives: By the end of this course Students will be able to explain the concepts behind computer based recognition and the extraction of features from raster images. Students will also be able to illustrate some successful applications of vision systems and will be able to identify the vision systems limitations.

Course Outline: Concepts behind computer-based recognition and extraction of features from raster images. applications of vision systems and their limitations. Overview of early, intermediate and high level vision, Segmentation: region splitting and merging; quadtree structures for segmentation; mean and variance pyramids; computing the first and second derivatives of images using the isotropic, Sobel and Laplacian operators; grouping edge points into straight lines by means of the Hough transform; limitations of the Hough transform; parameterisation of conic sections. Perceptual grouping: failure of the Hough transform; perceptual criteria; improved Hough transform with perceptual features; grouping line segments into curves. Overview of mammalian vision: experimental results of Hubel and Weisel; analogy to edge point detection and Hough transform; Relaxation labelling of images: detection of image features; Grouping of contours and straight lines into higher order features such as vertices and facets. Depth measurement in images.

Text Books/Reference Books:

1. Computer Vision: A Modern Approach, By David Forsyth, Jean Ponce, Prentice Hall, 2003.
2. Computer Vision, By Linda G. Shapiro, George C. Stockman, Prentice Hall, 2001.
3. Handbook of Mathematical Models in Computer Vision, By Nikos Paragios, Yunmei Chen, Olivier Faugeras, Birkhäuser, 2006.

Course Name: Rich Internet Applications

Course Structure: Lectures: 3 Labs: 0 **Credit Hours: 3**

Course Outline: This course covers the concept and technology evolution regarding the internet applications and the use of interface tools. Mainly, the course can focus on any one of the technologies of modern day, for example, macromedia's FLASH. However, the course will use the concepts of data structures, object oriented programming, programming languages and the software design and engineering to develop projects of medium to large magnitude.

Text Books/Reference Books: No particular text book can be specified as the contents and teaching approach depend on the instructor and the latest trends in the area. Macromedia's presence on the web can be utilized to maximum, however.

National Curriculum Revision Committee Software Engineering

The National Curriculum Revision Committee for Software Engineering (NCRC-SE) met on 24-26 November, 2008 to develop the vision for Software Engineering education and curriculum for software engineering programmes. The Committee met again on 7-9 April, 2009 to finalize the curricula recommendations. Participants represented most of the universities and software industry of the country. Following experts participated in the meetings:

- | | | |
|---|---|-----------------|
| 1 | Dr. S. M. Aqil Burney,
Meritorious Professor,
Chairman and Director,
Department of Computer Science,
University of Karachi. | Convener |
| 2 | Dr. Nazir A. Sangi
Professor,
Department of Computer Science,
Allama Iqbal Open University, Islamabad | Member |
| 3 | Dr. Imdad Ali Ismaili,
Professor and Director
Institute of Information and Communication Technologies
University of Sindh, Jamshoo. | Member |
| 4 | Dr. Zubair A. Shaikh
PEC Nominee
FAST National University of Computer & Emerging Sciences
National Highway | Member |
| 5 | Dr. Jerald Allan Kabell,
Professor & Chairman,
Department of Computer Science & I.T.,
Forman Christian College University, Lahore. | Member |
| 6 | Dr. Naveed Ikram,
Associate Professor,
Department of Computer Science,
International Islamic University,
H-10 Campus, Faculty Block-2, Islamabad | Member |
| 7 | Dr. Muhammad Abdul Qadir,
Professor and Dean, Faculty of Engineering & Applied Sciences,
Mohammad Ali Jinnah University,
Islamabad Campus, Jinnah Avenue, Islamabad
The Mall, Rawalpindi. | Member |

- | | | |
|----|---|---------------|
| 8 | Dr. Qamar Uddin Khand,
Associate Professor,
Department of Information Technology.
Sukkur IBA Airport Road, Sukkur | Member |
| 9 | Dr. Muhammad Akram Sheikh
Associate Professor,
Department of CS and SE Engineering,
Mehran University of Engineering & Technology, Jamshoro | Member |
| 10 | Dr. Najmi Ghani Haider
Professor and HOD
Department of Computing Science,
SZABIST, Karachi. | Member |
| 11 | Dr. Tahseen Ahmed Jilani,
Assistant Professor,
Department of Computer Science,
University of Karachi, Karachi | Member |
| 12 | Dr. Saeed Mahfooz, (NOT PRESENT)
Department of Computer Science,
University of Peshawar, Peshawar | Member |
| 13 | Mr. Ghulam Mujtaba Sheikh
Lecturer
IBA Sukkur Airport Road, Sukkur | Member |
| 14 | Dr. Shafay Shamail
Associate Professor and Head
Department of Computer Science
School of Science and Engineering
Lahore University of Management Science (LUMS),
Lahore. | Member |
| 15 | Mr. Arshad Iqbal,
Principal Engineer,
Department of Computer & Information Science,
P.O. PIEAS Nilore, Islamabad | Member |
| 16 | Dr. Shahid Nazir Bhatti,
Assistant Professor,
Department of Computer Science,
COMSATS Institute of Information Technology, Islamabad | Member |
| 17 | Mr. Nadeem Mahmood,
Assistant Professor,
Department of Computer Science, | Member |

- 18 Syed Asim Ali Shah, **(NOT PRESENT)**
Lecturer,
Bahria University,
Shangrila Road, Sector E-8, Islamabad

- 19 Abdul Mujeeb Kabadia, **(NOT PRESENT)**
Project Manager
NetSol Technologies Ltd,
NetSol Avenue, Ghazi Road,
Lahore Cantt., 54792

- 20 Dr. Zahoor Jan,
Assistant Professor,
Department of I.T. / Computer Science,
University of Peshawar,
Peshawar.

The Discipline of Software Engineering

Software Engineering is a bridge connecting the basic concepts and principles of Computer Science with the variety of users who can benefit from technologies based upon those principles. It includes the design and development of software systems which are effective, efficient, robust, maintainable, and maximally useful and usable. It also includes the design and development of techniques, processes and higher level tools by which these applications can be developed in a timely, cost effective and sustainable manner. At both levels it requires a systematic approach which deals with quantifiable measures of quality and effectiveness, as well as attention to the critical nature of the various products of the process.

Software engineering therefore requires familiarity with the basic needs and processes in the various application domains, with the principles of good engineering practice and with the underlying concepts and principles of computer science. It requires facility in problem analysis, solution design, program development and documentation. It also requires a basic understanding of the ways in which humans interact with technological systems.

A software engineering programme should develop professionals who have a mastery of software development principles, theory, practice, and process.

Software Engineering and Computer Science differ in much the same way as do Electrical Engineering and Physics¹. Generally, engineering should be concerned with applying what we already know to create products, while science is more theoretical. Therefore, the goal of Computer Science, according to Parnas², is to

¹ David Parnas, "Software Engineering Programmes are not Computer Science Programmes", *IEEE Software*, Nov/Dec. 1999, pp. 19-30.

² David Parnas, "Software Engineering Programmes are not Computer Science Programmes", *IEEE Software*, Nov/Dec. 1999, pp. 19-30.

learn and to extend the science. SE on the other hand aims to use the science and technology already available to create products and tools for use.

Software Engineering derives its essence from computer science as other engineering disciplines do from natural or life sciences, with an emphasis on issues of process, design, measurement, analysis and verification providing a strong foundation in engineering principles and practices as applied to software development.

Definition

Software Engineering is a discipline concerned with the development of software systems by applying engineering principles with the goal of developing cost-effective quality systems. There are many definitions in literature. Such as:

- "The establishment and use of sound engineering principles (methods) in order to obtain economically software that is reliable and works on real machines" [Bauer 1972].
- "Software engineering is that form of engineering that applies the principles of computer science and mathematics to achieving cost-effective solutions to software problems." [CMU/SEI-90-TR-003]
- "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software" [IEEE 1990].
- IEEE defines software engineering [IEEE-93] as
 1. The application of systematic, disciplined, quantifiable approach to development, operation, and maintenance of software; that is application of engineering to software.
 2. The study of approaches as in 1."

Software Engineering could also be defined as:

"The application of systematic, disciplined, quantifiable approach to design, development, deployment, and maintenance of reliable and economical software systems."

Vision

Software engineering is the discipline of creating high-quality software systems in a systematic, controlled and efficient manner. It involves the application of engineering concepts, techniques, and methods to the design, development, deployment and maintenance of software systems. A software engineering programme should develop professionals who have a mastery of principles, theory, practices, and processes necessary to produce quality software systems.. The curriculum committee formalized the Vision Statement for SE education in Pakistan as follows:

The SE education in Pakistan will focus on imparting the knowledge and training which should enable students to harmonize theory with practice, concept with application, and problem with solution. It will prepare them to apply ably engineering principles, practices, and processes to design, develop, deploy, and maintain software systems. The programme will lead to development of student's professional and interpersonal skills. It will help students to enhance their ability in oral and written communication, and their adaptability to team environments. The programme will inculcate among

students a strong sense of civic, professional and ethical responsibility. The programme will also strive to develop a capacity for innovation and a passion for life long learning.

SE curricula thus developed would reflect the aim to satisfy professional demands of the industry and academia both in terms of immediate needs and the capacity for longer term development. The graduates thus produced will be adequately equipped to exploit the opportunities and answer the challenges offered by the modern world.

Knowledge Areas of SE Curriculum Development ABET Engineering Criteria 2000 notes:

The curriculum must provide both breadth and depth across the range of engineering and computer science topics implied by the title and objective of the programme. The programme must demonstrate that graduates have: the ability to analyze, design, verify, validate, implement, apply, and maintain software systems; the ability to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer and management sciences to complex software systems.

SE curriculum specified here has been developed systematically by identifying the major knowledge areas of SE education, in the spirit of engineering criteria above. It is noted that efforts carried out by ACM and IEEE-CS to develop international software curricula are very relevant and provide excellent guidelines on the issue. Outcome of these efforts is documented in *Software Engineering Body of Knowledge (SWEBOK)*³, *Software Engineering Education Knowledge (SEEK)*⁴, and *Computing Curriculum 2008*⁵.

The following major areas of relevant pedagogy have been identified to be appropriate for design of the software engineering curriculum:

1. Computing Foundation (CS/SE/CE)
2. Software Engineering (SE Major)
3. Software Engineering Application Domain
4. Supporting Areas (Mathematics and Natural Sciences)
5. General Education (Management, Humanities, Social Sciences)

The committee is of the view that good curriculum should focus on building a solid foundation in the early stages of learning. It should gradually introduce and strengthen the core professional competencies and desired skill-sets. Software engineering concepts should be taken up as early as the start of 2nd year. The main technical SE content should be covered during the third and fourth years. Practical component should use medium to large scale projects to develop in students a systematic approach to problem solving and program development. Good SE practices must be nurtured all through the education programme. The practice of software engineering is often in the context of non-software application domains. The graduates, therefore, should be provided an opportunity for reasonably broad exposure to at least one application area in the senior years. It will help them learn and demonstrate the application of software engineering practices. A capstone design project should provide the opportunity to bring together all the knowledge

³ *Guide to Software Engineering Body of Knowledge*, 2004 Edition,.

⁴ *Software Engineering – Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, 2004 August 23, 2004

⁵ *Computing Curriculum 2008—Draft*

gained in a wide variety of courses to solve realistic problems in a team-based environment.

Software Engineering Degree Programs

Nomenclature

The committee emphasized that the nomenclature followed for Software Engineering programs should correspond to international trends and standards. The following nomenclature was thus agreed upon for various degrees:

Bachelor of Science in Software Engineering — *BS Software Engg* — *BS (SE)*

Master of Science in Software Engineering — *MS Software Engg* — *MS (SE)*

Duration of Programs

The committee defined a credit hour as 15 lecturing hours in a course offered in a particular semester. It was agreed that 3 weekly lab hours shall be treated as one credit hour for a course. In normal circumstances a semester comprises 15 teaching weeks followed by the final examination. The notation used for this purpose is X(Y-Z), X represents credit hours, y represents hours of practical in class per week and z represents hours of lab work per week over a 15 week semester.

The BS Software Engineering Degree would be a 4-year programme spread over 8 semesters and MS Software Engineering programme would be a 2-year programme spread over 4 semesters.

Admission Criteria

The eligibility criteria for BS Software Engineering admission was agreed to be intermediate with mathematics or equivalent qualifications, however, universities may define their own admission criteria.

The eligibility criterion for admission to MS Software Engineering was unanimously agreed to be 4-year BS Software Engineering or equivalent qualifications, however, universities may define their own admission criteria.

Curriculum for BS Software Engineering — BS (SE)

Curriculum Objective

The objective of the curriculum is to prepare students for professional careers and graduate studies with a balance between computing theory and practical application of software engineering concepts, methodologies, tools and technologies in the modern software development environments.

Graduates of such programs will be able to function as proficient software developers and effective team members. They will have grounding in communication, mathematics and science, and the cultural, historical, and social issues that influence and effect or relate to the development of high quality software systems. They will have knowledge of and experience with software product engineering and engineering management and an understanding of professional issues and practices. Graduates will be able to understand and assess their own software engineering capabilities and performance.⁶

The curriculum is designed to ensure breadth across allied disciplines and supporting subjects; and depth in most areas of the software engineering body of knowledge. Various components have been included in the curriculum to ensure that the graduates will:

- understand and be able to apply mathematics, physical science, computer science and related disciplines.
- understand and be able to apply the principles of software engineering practice and process, subject to realistic constraints.
- be able to model, analyze, document and track system requirements, both functional and non-functional.
- be able to design, implement, deploy and maintain software systems.
- be able to verify and validate the software systems.
- have an awareness of current industry standards and practices.
- be able to work in one or more application domains.
- understand and apply the principles of the team process.
- be able to understand and apply software project management skills: measurement, estimation, costing, planning, deployment and tracking of resources.
- have strong communication and interpersonal skills.
- be capable of independent learning.
- understand professional responsibility and application of ethical principles.
- have knowledge of economics, humanities and social sciences.

Curriculum Model

The curriculum is designed to achieve systematically the objectives set out above for the programme. It has been structured to suit the needs of the students, the demands of the market, and the trends of the industry. During the first two years of the programme the students will be given an underpinning in computer science, with special emphasis on software engineering — concepts, processes, and practices. The students will be exposed to the discipline in a systematic, gradual and definite way. Students will also be trained in the skills and techniques which are rooted in the basic sciences like mathematics and physics. These areas will be taken care of

⁶ While setting the objectives the committee benefited substantially from different universities' online documentation for similar programmes available on <http://sites.computer.org/ccse/SEprogrammes.html>.

in the supporting courses which have been allocated reasonably sufficient space. Students' personal traits and personality polishing will be cared for by the general education courses including communication and writing skills. A host of slots for elective courses have also been proposed to give to the students an opportunity to move towards their areas of interest.

During the senior years the students will be given exposure to the more specialised aspects of the discipline. They will also be given training in at least one application domain which will help institutions to prepare human resource well suited to the needs of different segments of the job market. In order to inculcate among them a scientific attitude they will go through a substantial lab work, which will prepare them for the industry and for further research oriented studies. The final year design project will mark the crystallisation and culmination of the students' four-year learning process. Figure SE1 illustrates the structure of the proposed curriculum, whereas the Table SE1 gives the credit hour distribution of the core and elective courses.

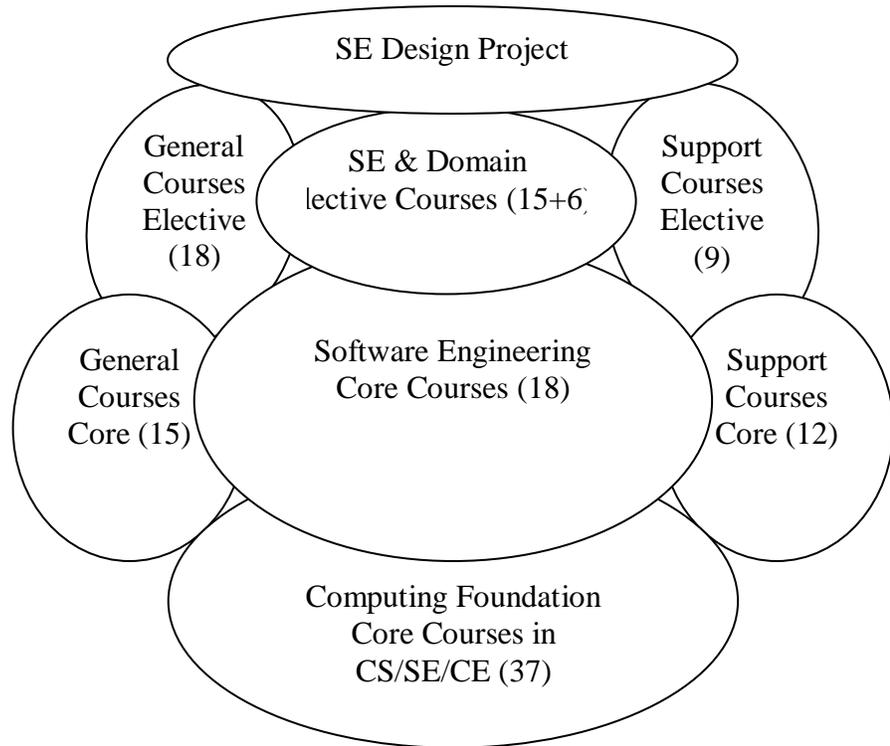


Figure-SE1 Structure of the proposed curriculum (credit hours within parenthesis)

Major Areas	Core/ Required	Electives	Credit Hours
Computing Foundation	43	21	82 (63%)
Software Engineering	18		
Software Engineering (Application Domain)	--	06	
Supporting Studies (Math/Science)	12	9	21 (17%)
General Education	15	12	27 (21%)
Total	82 (68%)	48 (32%)	130

Table SE1: The Credit Hour Distribution of the Core and Elective Courses

Computing-Core Courses – 43 Credit Hours

#	Co de	Pre- Req	Course Title	Credit hours	Proposed Semester
1		-	Introduction to Computing	4 (3-3)	1
2		-	Programming Fundamentals	4 (3-3)	1
3		1	Object Oriented Programming	3 (3-0)	2
4		-	Discrete Structures	3 (3-0)	2
5		3	Data Structure and Algorithms	3 (3-0)	3
6		3	Digital Logic and Design ¹	3 (3-0)	3
7		4	Operating Systems	4 (3-3)	4
8		4	Introduction to Database Systems	4 (3-3)	4
9		2	Introduction to Software Engineering ¹	3 (3-0)	4
10		6	Computer Communications and Networks ¹	3 (3-0)	6
11		8	Human Computer Interaction	3 (3-0)	7
12		-	Senior Design Project	6 (0-18)	7,8

1: Labs preferred in these courses. However, implementation details are left upon the concerned Institutes.

Bachelor of Science in Software Engineering: BS (SE)

Computing Core Requirements 37 Credit Hours (Refer to Computing part)

Required Software Engineering Courses (18/133)					
#	Code	Pre-req	Course Title	Credit Hours	Semester
12	SE	4	Software Construction	3 (2-3)	4
13	SE	7	Software Requirements Engineering	3 (2-3)	5
14	SE	13	Software Design and Architecture	3 (2-3)	6
15	SE	13	Software Quality Engineering	3 (2-3)	6
16	SE	7	Software Project Management	3 (3-0)	7
17	SE	7	Formal Methods in Software Engineering	3 (3-0)	5

Elective Computing & Software Engineering Courses (15/133)					
(The list below is by no means exhaustive. Institutions may add new course)					
#	Code	Pre-req	Course Title	Credit hours	Semester
12	SE	15	Software Metrics	3 (3,0)	7,8
13	SE	7	Software Engineering Economics	3 (3,0)	3-4
14	MG	-	Information System Audit	3 (3-0)	7
15	CS	7	Business Process Automation	3 (3-0)	7,8
16	CS	-	Design Patterns	3 (3-0)	7,8
17	SE	7	Software Testing	3 (2-3)	6,7
18	SE	3,4	Formal Methods	3 (3-0)	6-7
19	SE	7	PSP and TSP	3 (3-0)	7-8
20	SE	7	Distributed Computing	3 (3-0)	7-8
21	CS	30	Introduction to Soft Computing	3 (2-3)	4, 5
22	CS	7,10	Real-time systems	3 (3-0)	6-7
23	CS	8	Data Warehousing and Data Mining	3 (3,0)	6, 7
24	CS	3	Artificial Intelligence	3 (3-0)	6
25	CS	4	Data Security and Encryption	3 (3-0)	6
26	CS	3	Discrete Structures – II	3 (3-0)	2,3
27	CS	3,4	Automata Theory and Formal Languages	3 (3-0)	5, 6
28	CE	6	Microprocessor Interfacing	3 (3-0)	7
29	CS	4	Analysis of Algorithms	3 (3-0)	6
30	CS	2, 3	Principles of Programming Languages	3 (3-0)	7
31	CS	-	Computer Graphics	3 (2-3)	6
32	CS	30	Artificial Neural Networks	3 (3-0)	8
33	CS	8	Advance Database Management Systems	3 (2-3)	8
34	CS	-	Bio-Informatics	3 (3-0)	5-8
35	CS	-	Web-Engineering	3(3-0)	5-8

Domain Specific Elective Courses (6/133)

In-depth treatment of one of the following SE Application Domains should be offered in the form of set of two to three courses of 3 credits each in the selected domain. The list below is by no means exhaustive. Institutions may add new domains.

Each domain treatment should be organized as domain introduction, computing concept of the domains and the domain specific computing examples with general sprit of implementation using SE principles. Common domains may include banking, insurance, oil exploration; textile and garments; agriculture, medicine, defence, etc.

		Domains	Topics /Component	Cr	
1	IS	Enterprise Systems Engineering	ERP Systems, SCM Systems, CRM Systems	6	5-8
2	NS	Net-Centric Systems	Knowledge and skills in Web-based Technologies Depth in networking, Depth in security	6	5-8
3	IS	Enterprise Security Architecture	Business issues related to security, Security weaknesses and risk analysis, Cryptography, cryptanalysis, steganography, etc., Depth in networks	6	5-8
4	IS	Information Systems and Data Processing	Data warehousing, Depth in databases Depth in business administration	6	5-8
5	IS	Financial and E-commerce Systems	Accounting; Finance Depth in security	6	5-8
6	CE	Fault Tolerant and Survivable Systems	Knowledge and skills in heterogeneous, distributed systems; Depth in security, Intrusion detection Failure analysis and recovery	6	5-8
7	CE	Safety Critical Systems	Depth in formal methods, Proof of correctness, etc. Knowledge of control systems	6	5-8
8	CE	Embedded & Real time Systems	Hardware for embedded systems Languages and tools for development Depth in timing issues; Hardware verification	6	5-8
9	BI	Bio-medical Systems	Biology and related sciences Related safety critical systems knowledge	6	5-8
10	SS	Scientific Systems	Depth in related sciences; Depth in statistics Visualization and graphics	6	5-8
11	TE	Telecommunication Systems	Depth in signals, information theory, etc. Telephony and telecommunication protocols	6	5-8

12	AS	Avionic & Vehicular Systems	Mechanical engineering concepts Related safety critical systems knowledge Related embedded and real-time systems knowledge	6	5-8
14	IE	Industrial Process Systems	Control systems Industrial engineering and other relevant areas Related embedded and real-time systems knowledge	6	5-8
15	ES	Multimedia, game, and entertainment Systems	Visualization, haptics, and graphics Depth in human computer interface design Depth in networks	6	5-8
16	WN	System for Small & mobile Platforms	Depth in human computer interfaces for small and mobile platforms, Wireless technology Related embedded and real-time systems knowledge Related telecom systems knowledge	6	5-8
17	AI	Agent based Systems	Machine learning, Fuzzy logic Knowledge engineering	6	5-8

Computing Requirements-Supporting Sciences 12 Credit hours (refer to Computing part)

Elective Supporting Courses (9/133)					
<i>(The list below is by no means exhaustive. Institutions may add new course)</i>					
5.	MT	1	Advanced Calculus	3 (3-0)	2
6.	MT	3	Numerical and Symbolic Computing	3 (3-0)	5-6
7.	MT	3	Stochastic Processes	3 (3-0)	6-7
8.	Sc	-	Physics-II (Mechanics)	3(3-0)	2
9.	Sc	-	Bio-Chemistry	3 (3-0)	4
10.	Sc	-	Biology/ genetics	3 (3-0)	4
11.	EE	4	Digital Electronics	4 (3-3)	3-4
12.	Sc	--	Software Engineering Economics	3(3,3)	3-4
13.	MT	--	Computational Linear Algebra	3(3-0)	5-6
14.	MT	--	Mathematical tools for Software Engineering	3(3-0)	6-7
15.	MT	--	Operation Research	3(3-0)	5-6
16.	MT	--	Simulation and Modeling	3(3-0)	4-5
17.	CS	--	Natural Language Processing	3 (3-0)	6-7

Computing Requirements-General Education 15 Credit Hours (Refer to Computing part)

Elective General Education Courses (12/133)					
<i>(The list below is by no means exhaustive. Institutions may add new course)</i>					
60	SS	-	English Literature	3 (3-0)	5
61	SS	-	Economics	3 (3-0)	7
62	SS	-	Sociology	3 (3-0)	2-6
63	SS	-	Psychology	3 (3-0)	6
64	SS	-	International Relations	3 (3-0)	7
65	HU	-	Foreign Language (Arabic, French, German, etc.)	3 (3-0)	7-8
66	MG	-	Information System Audit	3 (3-0)	7
67	MG	-	Principles of Management	3 (3-0)	4
68	MG	-	Human Resource Management	3 (3-0)	5
69	MG	-	Marketing	3 (3-0)	6-7
70	MG	-	Accounting and Finance	3 (3-0)	5-7

Sample Scheme of Study for BS (SE)
4-year Programme (8 Semesters)
(130 Credit Hours)

Semester-wise 4-Year Plan			
Semester 1	Cr. Hrs.		Semester 2
Introduction to Computing	3		Discrete Structures-I
Programming Fundamentals	4		Object Oriented Programming
Calculus and Analytical Geometry	3		Supporting Elective I
Physics	3		GE/University Elective I
English-I (Functional English)	3		English-II (Communication Skills)
	16		
			15
Semester 3	Cr. Hrs.		Semester 4
Introduction to Software Engineering	3		Operating Systems
Data Structures and Algorithms	3		Software Construction
Digital Logic & Design	3		Supporting Elective II
Linear Algebra	3		GE/University Elective II
Pakistan Studies and Islamic Studies	3		Introduction to Database Systems
			English-III (Technical and Report Writing)
	15		
			18
Semester 5	Cr. Hrs.		Semester 6
Software Requirement Engineering	3		Human Computer Interaction
Probability and Statistics	3		Software Quality Engineering
Computer Communication and Networks	3		Software Design & Architecture
SE Elective I	3		Formal Methods in Software Engineering
Supporting Elective III	3		SE Elective II
GE/University Elective III	3		SE Application Domain Elective –I
	18		
			18
Semester 7	Cr. Hrs.		Semester 8
Senior Capstone Project I	3		Senior Capstone Project II
Software Project Management	3		SE Elective III
Professional Practice	3		SE Elective IV
SE Application Domain Elective –II	3		SE Elective V
GE/University Elective IV	3		
	18		
			12

COURSE CONTENTS

BS (SE) - Software Engineering

Course Name: Software Construction

Course Structure: Lectures: 2/ Labs: 1 Credit Hours: 3

Prerequisites: Data Structures and Algorithms

Objectives:

Upon completion of this course, students will have the ability to:

- Apply a wide variety of software construction techniques and tools, including state-based and table-driven approaches to low-level design of software
- Design simple languages and protocols suitable for a variety of applications
- Generate code for simple languages and protocols using suitable tools
- Create simple formal specifications of low-level software modules, check the validity of these specifications, and generate code from the specifications using appropriate tools
- Design simple concurrent software
- Analyze software to improve its efficiency, reliability, and maintainability

Course Outline:

- Basics of formal languages; syntax and semantics; grammars; Backus Naur Form. Parsing; regular expressions and their relationship to state diagrams
- Lexical Analysis; tokens; more regular expressions and transition networks; principles of scanners
- Using tools to generate scanners; applications of scanners. Relation of scanners and compilers
- Parsing concepts; parse trees; context free grammars, LL Parsing
- Overview of principles of programming languages. Criteria for selecting programming languages and platforms
- Tools for automating software design and construction. Modelling system behaviour with extended finite state machines
- SDL
- Representing concurrency, and analyzing concurrent designs

Lab Work:

- Use of software engineering tools to create designs
- Use of parser generators to generate languages

Reference Material:

1. Object-Oriented Software Construction, by Bertrand Meyer, Second Edition, Published by, Prentice Hall in 1997
2. Formal Methods in Computing by M. Ferenczi, and Andras Pataticza , Sep 2005
3. Code Complete 2nd edition: A practical handbook of software construction, published by Microsoft Press, 2004.
4. Software Engineering by Ian Sommerville, 8th edition, Addison & Wesley. 2006

Course Name: Software Requirement Engineering

Course Structure: Lectures: 2 / Labs: 1 Credit Hours: 3

Prerequisites: Introduction to Software Engineering

Objectives: To understand Issues in Requirements Engineering, to understand and apply Requirements Engineering Process, to understand and use Requirements Elicitation and Specification, to understand and use Formal Techniques, to understand modeling and analysis of Non-Functional Requirements.

Course Outline: Definition of requirements engineering and role in system development, Fundamental concepts and activities of requirements engineering, Information elicitation techniques, Modeling scenarios

Fundamentals of goal-oriented requirements engineering, Modeling behavioral goals, Modeling quality goals, Goal modeling heuristics, Object modeling for requirements engineering, Object modeling notations, Object modeling heuristics, Identifying objects from goals, Modeling use cases and state machines, Deriving operational requirements from goals, Requirements Specification, Requirements verification and validation

Management of inconsistency and conflict, requirements engineering risks, the role of quality goals in the requirements selection process, Techniques for requirements evaluation, selection and prioritization; Requirements management; Requirements traceability and impact analysis.

Lab Work

- The requirements are for the development of case applications.
- Outline of business requirements, the product vision and scope for applications.
- Use case elicitation using automated tools e.g. UML, Mobile Scenarios and PDA's etc
- Development of Software Requirement Specification (SRS)
- Requirement Engineering Group Discussion activity and resource allocation etc

Reference Material:

1. Hull, Jackson, and Dick, Requirements Engineering, 2004, Springer
2. Karl E. Wiegers, Software Requirements, 2nd Edition, 2003, Microsoft Press
3. Loucopoulos and Karakostas, System Requirements Engineering, McGraw-Hill , 1995
4. Kotonya and Sommerville, Requirements Engineering: Processes and Techniques, John Wiley Sons, 1998.

Course Name: Software Design and Architecture

Course Structure: Lectures: 2 / Labs: 3 Credit Hours: 3

Prerequisites: Software Requirement Engineering

Objectives: An in-depth look at software design. Continuation of the study of design patterns, frameworks, and architectures. Survey of current middleware architectures. Design of distributed systems using middleware. Component based design. Measurement theory and appropriate use of metrics in design. Designing for qualities such as performance, safety, security, reusability, reliability, etc. Measuring internal qualities and complexity of software. Evaluation and evolution of designs. Basics of software evolution, reengineering, and reverse engineering.

Upon completion of this course, students will have the ability to:

- Apply a wide variety of design patterns, frameworks, and architectures in designing a wide variety of software
- Design and implement software using several different middleware technologies
- Use sound quality metrics as objectives for designs, and then measure and assess designs to ensure the objectives have been met
- Modify designs using sound change control approaches
- Use reverse engineering techniques to recapture the design of software

Course Outline:

Introduction: Putting Software Architecture in Context, Software Architecture as a Design Plan, Software Architecture as an Abstraction, Software Architecture Terminology, Four Views of Software Architecture, Loose coupling between Views, Engineering concerns addressed by different views, Using the four Views.

Global Analysis: Overview of Global Analysis Activities. Analyze Factors. Develop Strategies. Analyze Organizational Factors. Begin Developing Strategies. Analyze Technological Factors. Continue Developing Strategies. Analyze Product Factors. Continue Developing Strategies.

Conceptual Architecture View: Design Activities for the Conceptual Architecture View. Central Design Tasks: Components, Connectors, and Configuration. Final Design Task: Resource Budgeting, Traceability, Uses for the Conceptual Architecture View.

Module Architecture View: Design Activities for the Module Architecture View. Central Design Tasks: Modularization and Layering, Final Design Task: Interface Design, Traceability, Uses for the Module Architecture View.

Execution Architecture View: Design Activities for the Execution Architecture View. Central Design Tasks: Runtime Entities, Communication Paths, and Configuration, Final Design Task: Resource Allocation, Traceability, Uses for the Execution Architecture View.

Code Architecture View: Design Activities for the Code Architecture View. Central Design Tasks, Final Design Tasks, Traceability, Uses for the Code Architecture View.

Role of Architect: The Architect as a Key Technical Consultant, The Architect Makes Decisions, The Architect Coaches, The Architect Coordinates, The Architect Implements, The Architect Advocates, Software Architecture as a Career.

Lab Work

- Planning and Practice of existing software design methodologies.
- Outline of requirements, the existing design & architecture practices using up to date tools and technologies
- Applications of Software Development Life Cycle (SDLC), its phases and thus implementation of different process models
- Efficient use of different modeling and design tools e.g. UML (for code generation), open source code development etc.
- Writing of Software Design Specifications

Resources:

1. Software Architecture Design - Methodology and Styles Stipes Publishing L.L.C. Copyright © 2006 Lixin Tao, Xiang Fu and Kai Qian
2. Christine Hofmeister, Robert Nord, Dilip Soni, Applied Software Architecture, 1999, Pearson.

Course Name: **Software Quality Engineering**

Course Structure: Lectures: 2 /Labs: 1 Credit Hours: 3

Prerequisites: Software Requirement Engineering

Objectives: The objective of this course is to make students have ability to understand and practice:

How to assure and verify Quality, and the need for a culture of quality. Avoidance of errors and other quality problems. Inspections and reviews. Testing, verification and validation techniques using variety of tools, incorporation of feedback loop to support quality promotion. Process assurance vs. Product assurance. Quality process standards. Product and process assurance. Problem analysis and reporting. Statistical approaches to quality control.

Course Outline: Introduction to software quality assurance, The Quality Challenge, Quality Control v/s Quality Assurance, Quality Assurance in Software Projects (Phases), Principles and Practices, Quality Management, Quality Assurance and Standards, Quality Planning and Quality Control, Verification and Validation, Planning Verification and Validation, Critical System Validation, Reliability Validation, Safety Assurance, Security assessment, Inspections and reviews, Principles of software validation, Software verification, Planning for Software Quality Assurance, Software Quality Assurance (SQA) Plans, SQA-Organizational Level Initiatives, SQA Planning (Observations, Numbers, Results), Software Testing, Specification based test construction techniques, White-box and grey-box testing, Others comprehensive software testing techniques for SDLC, Control flow oriented test construction techniques, Data flow oriented test construction techniques, Clean-room approach to quality assurance, Product Quality and Process Quality, Standards for process quality and standards for product quality, Walkthroughs and Inspections, Structure, Checklist, Audits, Roles and Responsibilities (Reviews, Inspections, etc), How to make Reviews and Inspections most effective

Lab Work:

- Planning and Development of test cases
- Planning and implementation of different Testing Techniques e.g. White Box Testing, Black Box Testing, Recursion Testing etc

- Collection and Generation of test data
- Practicing Testing methodologies using automated testing tool & technologies
- Analysis of Test results & Extreme testing

Resources:

1. Software Quality Assurance: Principles and Practice (Hardcover). by Nina S. Godbole, published by Alpha Science, 2004
2. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement by Jeff Tian, published by John Wiley & sons, 2005
3. Software Testing in the Real World: Improving the Process by Kit, Edward, Addison & Wesley, 1998.
4. Perfect Software: And other illusions about testing by Gerald M. Weinberg, published Dorest House, 2008

Course Name: Software Project Management

Course Structure: Lectures: 3 /Labs: 0 Credit Hours: 3

Prerequisites: Introduction to Software Engineering

Objectives: To develop ability to plan and manage software development projects successfully, maximizing the return from each stage of the software development life cycle.

Course Outline: Software Crisis and Software Engineering, Classic Mistakes, Overview of Project Management, PMI Process Groups, Software project Phases, Project charter, Statement of Work (SOW), Planning Phase: Development lifecycle models, matching lifecycles to projects, Project plans, Work Breakdown Structures (WBS), Estimation of effort and cost (Expert Judgment, FP and Use Case point methods), Scheduling: Project network diagram fundamentals, CPM, PERT, Gantt charts, Critical chain scheduling, Using MS-Project, Assigning Resources, Resource leveling, Team models, Managing conflict and motivating, Project Monitoring and Control: Status reporting, Project metrics, EVM, Communications Techniques, Risk management and Change control Project Recovery, Documentation, Cutover/Migration, Post Project Reviews, Closing.

Resources

1. Bob Hughes and Mike Cotterell , Software Project Management, 2005, McGraw Hill Higher Education
2. Dwayne Phillips. The Software Project Manager's Handbook - Principles that work at work. 2nd Edition, IEEE Computer Society Press and Wiley Interscience, 2004. ISBN 0-471-67420-6

BS (SE) - Software Engineering Courses (Electives)

Course Name: Software Metrics

Course Structure: Lectures: 3 Lab:0 Credit Hours: 3

Prerequisites: Software Quality Engineering (SQE)

Objectives:

Upon completion of this course, students will have the ability to:

- Take account of the Metrics Program.
 - Measurement theory (overview of software metrics, basics of measurement theory, goal-based framework for software measurement, empirical investigation in software engineering)
 - Identify the internal and external metrics attributes
 - Enhancing the software development process with respect to metrics
 - Software product and process measurements (measuring internal product attributes: size and structure, measuring external product attributes: quality, measuring cost and effort, measuring software reliability, software test metrics, object-oriented metrics)
 - Measurement management
- Account of well known International metrics in software and system engineering

Course Outline:

- What are software metrics, Basic Measurement Theory
- Measurement quality, Measurement process, Measurement validation
- Software measure classification
- Goal-based paradigms: Goal-Question-Metrics (GQM), Goal-Question-Indicator-Metrics (GQIM) and Applications of GQM and GQIM
- Design Metrics, Measurements and Models, Measurements Scales
- Software engineering investigation, Investigation principles, Investigation techniques, Formal experiments: Planning, Formal experiments: Principles and Formal experiments: Selection
- Internal Metrics, Types of metrics, Software Size, Software Size: Length (code, specification, design), Software Size: Reuse, Software Size: Functionality (function point, feature point, object point, use-case point), Software Size: Complexity
- Representing concurrency, and analyzing concurrent designs
- Software structural measurement, Control-flow structure, Cyclomatic complexity, Data flow and data structure attributes, Architectural measurement
- Software cost model, COCOMO and COCOMO II, Constraint model, Software Lifecycle Management (SLIM), Cost models: advantages and drawbacks
- Software quality, Software quality models: Boehm's model, McCall's model, ISO 9126 model, Especially account of ISO/ IEC 9126 External Metrics suite etc., Basic software quality metrics, Quality management models, Measuring customer satisfaction
- Object-Oriented measurement concepts, Basic metrics for OO systems, CK metrics, OO analysis and design metrics, Metrics for productivity measurement, Metrics for OO software quality
- SQA, Test concepts, definitions and techniques, Estimating number of test case, Allocating test times , Decisions based on testing, Test coverage measurement, Software testability measurement, Remaining defects measurement

Lab Work:

- Use of software engineering tools to estimate attributes of existing well known metrics
- Estimate the attributes and sub-attributes of the SDLC depending upon the assigned data/project
- Applying ISO external metrics attributes to existing SDLC phases

Reference Material:

Additional Recommended Text and Reference Books:

1. Metrics and Models in Software Quality Engineering, by Stephen H. Kan, 2nd ed. Addison-Wesley Professional (2002)
2. Software Metrics: A Rigorous and Practical Approach, (2nd ed.), by N.E. Fenton and S.L. Pfleeger, PWS Publishing, 1998
3. Software Engineer's Reference Book, by J. McDermid (Edt.), Butterworth Heinemann. Year of Publication
4. Software Metrics: A Guide to Planning, Analysis, and Application, C. Ravindranath Pandian, Auerbach Publications, (2004).
5. Applied Software Measurement: Assuring Productivity and Quality, C. Jones, McGraw-Hill. Year of Publication
6. ISO/IEC 9126 External Metrics Reports I & II
7. Guide to Advance Empirical Software Engineering by Forrest Shull, Janice Singer (Eds.), Springer-Verlag, 2007.

Course Name: **Software Engineering Economics**

Course Structure: Lectures: /Labs: Credit Hours: 3

Prerequisites:

Objectives: Determine how new software development technologies affect the economics and risks of software development. Understand and characterize how the paradigm shift affects or replaces our current methods of software cost, schedule and risk estimation. Identify best practices and lessons learned with Web-based developments. Identify acquisition and lifecycle risks

Course Outline: Programming aspects, economic aspects, human relations aspects, software trends: cost, social impact, the plurality of SE Means, The GOALS Approach to Software Engineering, The Software Work Breakdown Structure (WBS), Software Maintenance, introduction to COCOMO, definitions and assumptions, development effort and schedule, phase distribution, The Rayleigh Distribution, interpolation, basic software maintenance effort estimation. Performance Models, Optimal Performance, Sensitivity Analysis, Cost-Effectiveness Models.

Resources:

1. Boehm, Software Engineering Economics, Prentice Hall, 1981.
2. Boehm et al., Software Cost Estimation with COCOMO II , Prentice Hall, 2000.
3. Reifer, Don. Making the Software Business Case: Improvement by the Numbers , Addison Wesley, 2001.

Course Name: **Information System Audit**

Course Structure: Lectures:3 / Labs: 0 Credit Hours: 3

Prerequisites: None

Objectives: To provide basic concept of information system audit and control, policies and procedures as defined by ISACA. To review and evaluate or conduct IS audits of an organization

Course Outline: IS Audit charter, Policies, Procedures, Audit computer networks and communication, Auditing software development, Acquisition, Maintenance, Auditing IT infrastructure, Auditing Management and Organization, Business process re engineering: IS audit proposal, report, evidence and follow-up, complaint to standard, Enterprise service agreement, IP pro count policies and process, Backup and procedures

Resources:

1. Auditing Information Systems, by Jack J. Champlain, published by John Wiley & Sons, 2003
2. *Control Objective for Information Technology (COBIT)*, 3rd Ed, by Information System Audit and Control Foundation.
3. *CISA Review Manual, 2004*, by Information System Audit and Control Association, www.isaca.org.

Course Name: Business Process Automation

Course Structure: Lectures:3/Labs:0 Credit Hours: 3

Prerequisites: Introduction to Software Engineering

Objectives:

Upon completion of this course, students will have the ability to:

- Apply their knowledge of business processes in the development of applications for various industry verticals
- Analyze business processes in terms of rules, complexity, user interactions and bottlenecks
- understand the concepts, views and latest methodologies of business process modeling
- utilize software tools for business process designing
- understand key concepts in the design and utilization of best business practices embedded in large business applications (ERP)
- have an appreciation of issues pertaining to organizational design and organizational change management in the context of business process management

Course Outline:

- Business Process Definitions
- Business Process Analysis and Modelling
- Business Process Lifecycle
- Policies, Procedures and Rules (in terms of business processes)
- Role of People, Customers, Trading Partners and Suppliers in Business Processes
- Business Process Simulation
- Business Process Re-Engineering (objectives and techniques)
- Basic concepts of Six Sigma (in terms of business process improvement)

Reference Material:

1. Business Process Automation ARIS in Practice by August Wilhelm Scheer, Springer-Verlag, 2004
2. Business Process Automation, 'Performance and Capacity Planning with Bpel' by Matthies Masour, Andy Scherzinger, VDM-Verlag 2007.

Course Name: Design Patterns

Course Structure: Lectures: 3 /Labs: 0 Credit Hours: 3

Prerequisites:

Objective: This course provides good knowledge about design patterns and how they are practically implemented in order to enhance existing systems and their design solutions.

After the course you will have a deep understanding regarding the thoughts behind design patterns. You will also have a knowledge database consisting of usable design patterns and related concepts, which will make you well-prepared for implementation in your daily work.

Course Outline: The course focuses on studying a large number of general design patterns and their practical application.

Furthermore, some patterns and idioms (language specific techniques) meant for real-time systems will be provided.

The course may include following contents:

General design patterns, Specific patterns for technical real-time systems, Deep understanding of the thoughts behind design patterns, Classification of patterns, Orientation around other types of patterns.

Resources:

1. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Third Edition
2. by Craig Larman, published by Prentice hall, 2004
3. Design Patterns Explained: A New Perspective on Object-Oriented Design, 2/e by James Trott (Kindle Edition - Feb 24, 2009)

Course Name: Software Testing

Course Structure: Lectures: 2 / Labs: 1 Credit Hours: 3

Prerequisites: Software Construction

Objectives: Testing techniques and principles: Defects vs. failures, equivalence classes, boundary testing.

Types of defects. Black-box Vs. Structural testing. Testing strategies: Unit testing, integration testing, profiling, test driven development. State based testing; configuration testing; compatibility testing; web site testing. Alpha, beta, and acceptance testing. Coverage criteria. Test instrumentation and tools. Developing test plans. Managing the testing process. Problem reporting, tracking, and analysis.

Learning objectives:

Upon completion of this course, students will have the ability to:

- Analyze requirements to determine appropriate testing strategies.
- Design and implement comprehensive test plans
- Apply a wide variety of testing techniques in an effective and efficient manner
- Compute test coverage and yield according to a variety of criteria
- Use statistical techniques to evaluate the defect density and the likelihood of faults.

- Conduct reviews and inspections.

Lab Work

Additional teaching considerations:

This course is intended to be 95% testing, with deep coverage of a wide variety of testing techniques.

The course should build skill and experience in the student, preferably with production code.

Course Outline: Introduction and overview: Testing and inspection concepts, Testing categories, Inception process: Objective of formal inspection Organizing Test cases: Decision Tables, Black box and white box testing Unit testing, Integration testing, Regression testing, System testing, user acceptance testing, Metrics and complexity, State based testing, Syntax testing; Use of software testing tools.

Resources:

1. *Introduction to Software Testing* by Paul Ammann and Jeff Offutt Published February 2008. , Cambridge University Press, Cambridge, UK.
2. *Software Testing* by Ron Patton, 2nd edition, SAMS publishing, 2005
3. *Software Testing in the Real World: Improving the Process* by Kit, Edward. (1995)

Course Name: Formal Methods

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives: In this course students learn how to represent computing systems with both state-based and process algebra models. They specify computing systems formally, reason about specifications, and verify their properties. They connect specifications to programmes through refinement and decomposition. They use theorem proving and model checking tools.

Course Outline: Introduction to formal specification, Transformational development, Specification analysis and proof, Programme verification, Objects and types: Sets and set types, Tuples and Cartesian product types, Bindings and schema types, Relations and functions, Properties and schemas, Generic constructions, The Z Language, Syntactic conventions, Schema references, Schema texts, Predicates, Schema expressions, Generics, Sequential Systems.

Resources:

1. *System Development using VDM* by Jones, C. B. (Year of Publication)
2. *Z – Specification Language* by Spivey (Year of Publication)
3. *Modern Formal Methods and Applications* by Hossam A. Gabbar, Springer-Verlag 2006.

Course Name: Distributed Computing

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: Introduction to Software Development

Objectives: This course is intended to provide a sound background for net centric software development. The course will concentrate an overview of major technologies like CORBA, RMI, .NET and will highlight the interfacing of middle layer with the upper layers and system layer

Course Outline: Introduction to distributed systems, Distributed data, Distributed processing system, Multithreading, Thread synchronization, Resource brokerage, Resource monitoring, Load balancing, Storage elements, Batch processing models, Middle layer architecture, Resource clustering, RMI, CORBA, Net, MPI.

Resources:

1. Distributed Systems: Principles and Paradigms by Tanen Baum. 2nd Edition
2. Distributed Systems: Concepts and Design (International Computer Science Series) by Jean Dollimore, Tim Kindberg, and George Coulouris (Hardcover - 14 Jun 2005)

Course Name: Introduction to Soft Computing

Course Structure: Lectures: 3/ Labs: 0 Credit Hours: 3

Prerequisites: Artificial Intelligence

Objectives: The course provides an in-depth overview of the theoretical and the practical aspects of the soft computing paradigm. The main focus is on the theory and application of probabilistic graphical models (commonly known as Bayesian networks in the Artificial Intelligence community) and related topics, such as, knowledge elicitation issues, belief updating in singly and multiply connected networks, simulation schemes for belief updating, parameter and structure learning of Bayesian networks, and integration of time and uncertainty. Alternative models of uncertain reasoning (including belief function theory and fuzzy logic) and biologically inspired computational models (neural networks and evolutionary algorithms) are also presented.

Introduction: Introduction to soft computing; introduction to biological and artificial neural network; introduction to fuzzy sets and fuzzy logic systems.

Biological neural networks: generalization of single neuron; neural dynamics; additive and shunting neural networks; short term and long-term memory. (Week 2-4)

Artificial neural networks and applications: artificial neural network models; learning in artificial neural networks; neural network applications in control systems. (Week 5-8)

Fuzzy systems and applications: fuzzy sets; fuzzy reasoning; fuzzy inference systems; fuzzy control; applications of fuzzy systems. (Week 9-11)

Course Outline:

1. Neural Networks
2. Classification Tree
3. Naïve Bayes
4. Applications of Predictive Models
5. Probabilistic Reasoning using Bayesian Networks
 - a. Knowledge Acquisition
 - b. Belief Updating
 - c. Exact and Simulation-based Propagation Algorithms

- d. Parameter and Structure Learning
- 6. Influence Nets
- 7. Dempster-Shafter Theory of Belief Functions
- 8. Fuzzy Logic

Resources:

1. Kevin Korb and Ann Nicholson, *Bayesian Artificial Intelligence*, 2003.
2. Ben Coppin, *Artificial Intelligence Illuminated*, 2004.
3. *Soft Computing & Intelligent Systems Design*, by Karray & De Silva, Addison-Wesley, 2005.

Course Name: Data Warehousing and Data Mining

Course Structure: Lectures: 3 Lab: 0 Credit Hours: 3

Course Outline: Concepts of Data mining and Data Warehousing, Data Preparation Techniques: outlier and missing data analysis, Data Reduction Techniques, learning methods in Data mining, Statistical Methods in Data Mining, Cluster Analysis, hierarchical, agglomerative and Naïve Bayesian methods, Decision Trees and Decision Rules, Association Rules, Other Soft Computing Approaches in Data Mining, Artificial Neural Networks, Fuzzy Logic and Fuzzy Set Theory, Genetic Algorithm, evolutionary algorithms.

Text Books/ References Books

1. Mehmed Kantatardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, 2003, John Wiley and Sons.
2. Margaret H. Dunham and S. Sridhar, *Data Mining, Introductory and Advanced Topics*, 2006, Pearson Education,
3. David Hand, Heikki Mannila and Padhraic Smyth, *Principles of Data Mining*, 2001, The MIT Press.
4. Daniel T. Larose, *Data Mining Methods and Models*, 2006, John Wiley and Sons.
5. Max Bramer, *Principles of Data Mining*, 2007, Springer-Verlag.
6. Paulraj Ponniah, *Data Warehousing Fundamentals*, 2005, John Wiley and Sons.
7. *Chuck Ballard Dirk Herreman Don Schau Rhonda Bell, Eunsang Kim Ann Valencic, Data Modeling Techniques for Data Warehousing*, 1999, IBM Corporation, International Technical Support Organization.

Course Name: Artificial Intelligence

Course Structure: Lectures: 2 / Labs: 3 **Credit Hours:** 3

Prerequisites: Data Structures

Objectives: This course focuses on the set of computational tools and techniques, which mimic the human decision-making process and capability.

Course Outline: Introduction to Common Lisp. AI classical systems: General Problem Solver, rules, simple search, means-ends analysis. ELIZA, pattern matching, rule based translators, OPS-5. Knowledge Representation: Natural language, rules, productions, predicate logic, semantic networks, frames, objects, scripts. Search: Depth first search, breadth first search, best first search, hill climbing, min-max search, A* search. Symbolic Mathematics: student, solving

algebra problems, translating English equations, solving algebraic equations, simplification rules, re-write rules, meta-rules, Macsyma, PRESS, ATLAS. Logic Programming: Resolution, unification, horn-clause logic, Prolog, Prolog programming. Sample case studies of shells and Knowledge Based Systems. A brief appreciation of state of the art computational techniques like neural networks, genetic algorithm, fuzzy sets.

Reference Material:

1. *Artificial Intelligence* by Luger, 4th edition Pearson Education.
2. Russell and Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed, Pearson Education.

Course Name: **Data Security and Encryption**

Course Structure: Lectures: 3/Labs: 0 Credit Hours: 3

Prerequisites:

Objectives: This is an introductory course on the methods, algorithms, techniques, and tools of data security and cryptography. After studying the theoretical aspects of cryptographic algorithms and protocols, we show how these techniques can be integrated to solve particular data and communication security problems. This course material is of use to computer and communication engineers who are interested in embedding security into an information system, and thus, providing integrity, confidentiality, and authenticity of the documents and the communicating parties.

Course Outline: The course consists of three parts: mathematical background, cryptography, and network security. The first part (mathematical background) introduces the principle of number theory and some results from probability theory, including Primes, random numbers, modular arithmetic and discrete logarithms. The second part (cryptography) covers cryptographic algorithms and design principles, including conventional and symmetric encryption (DES, IDEA, Blowfish, Rijndael, RC-4, RC-5), public key or asymmetric encryption (RSA, Diffie-Hellman), key management, hash functions (MD5, SHA-1, RIPEMD-160, HMAC), digital signatures, and certificates. The third part (network security) deals with practical applications that have been implemented and are in use to provide network security, including authentication protocols (X.509, Kerberos), electronic mail security (S/MIME, PGP), web security and protocols for secure electronic commerce (IPSec, SSL, TLS, SET).

Resources:

1. William Stallings, *Cryptography and Network Security: Principles and Practice*, 4th edition, Prentice Hall, 2005

Course Name: **Discrete Structures–II**

Course Structure: Lectures: 3 / Labs: 0 Credit Hours: 3

Prerequisites: None

Objectives: Continues the discussion of discrete mathematics introduced in CS105. Topics in the second course include predicate logic, recurrence relations, graphs, trees, matrices, computational complexity, elementary computability, and discrete probability.

Course Outline:

- Review of previous course
- Predicate logic: Universal and existential quantification; modus ponens and modus tollens; limitations of predicate logic
- Recurrence relations: Basic formulae; elementary solution techniques
- Graphs and trees: Fundamental definitions; simple algorithms; traversal strategies; proof techniques; spanning trees; applications
- Matrices: Basic properties; applications
- Computational complexity: Order analysis; standard complexity classes
- Elementary computability: Countability and uncountability; diagonalization proof to show uncountability of the reals; definition of the P and NP classes; simple demonstration of the halting problem
- Discrete probability: Finite probability spaces; conditional probability, independence
- Methods of Proof, Mathematical Induction and Recursion, loop invariants, Pigeon hole principle, Trees and Graphs, Optimization and matching.

Resources:

1. *Discrete Mathematical Structures* by Rosen 2006.
2. *Discrete Mathematics* by Richard Johnsonbaugh 1996.

Course Name: Theory of Automata and Formal Languages

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: Discrete Structures

Objectives: The course aims to develop an appreciation of the theoretical foundations of computer science through study of mathematical & abstract models of computers and the theory of formal languages. *Theory of formal languages* and use of various abstract machines as ‘recognizers’ and parsing will be studied for identifying/validating the synthetic characteristics of programming languages. Some of the abstract machines shall also study as “Transducers”.

Course Outline: *Finite State Models:* Language definitions preliminaries, Regular expressions/Regular languages, Finite automata (Fas), Transition graphs (TGs), NFAs, Kleene’s theorem, Transducers (automata with output), Pumping lemma and non regular language *Grammars and PDA:* Context free grammars, Derivations, derivation trees and ambiguity, Simplifying CFLs, Normal form grammars and parsing, Push-down Automata, Pumping lemma and non-context free languages, Decidability, Chomsky’s hierarchy of grammars *Turing Machines Theory:* Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Context sensitive Grammars, Defining Computers by TMs.

Resources:

1. *Automata Theory by Martin*
2. *Introduction to Computer Theory*, by Denial Cohen, John Wiley & Sons, Inc., 1996
3. *Introduction to Automata Theory, Languages and Computation*, by J Hopcraft, D. Ullman, 2006.
4. *Languages and Machines, An Into to the Theory of Comp. Sc.*, by Thomas A. Sudkamp.

Course Name: Analysis of Algorithms

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: Discrete Structures, Data Structures

Objectives: Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis on the structure, complexity, and efficiency of algorithms.

Course Outline: Introduction; Asymptotic notations; Recursion and recurrence relations; Divide-and-conquer approach; Sorting; Search trees; Heaps; Hashing; Greedy approach; Dynamic programming; Graph algorithms; Shortest paths; Network flow; Disjoint Sets; Polynomial and matrix calculations; String matching; NP complete problems; Approximation algorithms.

Resources:

1. *Introduction to Algorithms, Second Edition*, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, MIT press, 2001

Course Name: Computer Graphics

Course Structure: Lectures: 2 / Labs: 1 **Credit Hours:** 3

Prerequisites: Object Oriented Programming

Objectives: Study of various algorithms in computer graphics and their implementation in any programming language.

Course Outline: Graphics hardware. Fundamental algorithms. Applications of graphics. Interactive graphics programming - graph plotting, windows and clipping, and segmentation. Programming raster display systems, panning and zooming. Raster algorithms and software - Scan-Converting lines, characters and circles. Region filling and clipping. Two and three dimensional imaging geometry and transformations. Curve and surface design, rendering, shading, colour, and animation.

Resources:

1. *Computer Graphics Using Open GL, 2/E*, Francis S. Hill, Jr., Prentice Hall, 2001
2. *Fundamentals of Computer Graphics: 2nd Edition* by Peter Shirley A.K. Peters, 2005

Course Name: Artificial Neural Networks

Course Structure: Lectures:3/ Labs: Credit Hours: 3

Prerequisites: Artificial Intelligence

Objectives: This course presents an overview of the theory and applications of artificial neural network and fuzzy systems to engineering applications with emphasis on signal processing and control. The objective of this course is on the understanding of various neural network and fuzzy systems models and the applications of these models to solve engineering problems.

Course Outline: Introduction Contexts for and Motivation Neural Networks: Artificial Intelligence | Biological | Physics, Artificial Neural Network overview.

Supervised Learning: Single-Layer Networks , Perceptrons , Adalines Supervised Learning: Multi-Layer Networks.

Multi-Layer Perceptrons (MLPs) , Backpropagation , Conjugate Gradient method , Levenberg-Marquardt (LM) method , Madalines , Radial-Basis Networks , Cascade-Correlation Networks , Polynomial Networks , Recurrent Networks (Time series , Backpropagation through time , Finite Impulse Response (FIR) MLP), Temporal Differences method (TD).

Unsupervised Learning

Simple Competitive Networks: Winner-take-all | Hamming network , Learning Vector Quantization (LVQ), Counterpropagation Networks (CPN) , Adaptive Resonance Theory (ART) , Kohonen Self-Organizing Maps (SOMs) , Principal Component Analysis networks (PCA)

Associative Models

Linear Associative Memory (LAM) , Hopfield Networks , Brain-State-in-a-Box , BSB) , Boltzmann Machines and Simulated Annealing , Bi-Directional Associative Memory (BAM)

Optimization Problems

Neural Network Approaches, Evolutionary Programming , Fuzzy logic and its connection to NNs

Resources:

1. Neural Networks: A Comprehensive Foundation, Simon Haykin, Prentice Hall, Upper Saddle River, NJ, SECOND EDITION, 1999
2. Artificial neural networks: an introduction, by Kevin L. Priddy, Paul E. Keller- Technology & Engineering-2005
3. Neural networks: methodology and applications, by G. Dreyfus-computers-, 2005

Course Name: Bioinformatics

Course Structure: Lectures: 3/ Labs: 0 Credit Hours: 3

Prerequisites:

Objectives: This course introduces the scientist to Bioinformatics, which uses computer databases to store, retrieve and assist in understanding biological information. Genome-scale sequencing projects have led to an explosion of genetic

sequences available for automated analysis. These gene sequences are the codes, which direct the production of proteins that in turn regulate all life processes. The student will be shown how these sequences can lead to a much fuller understanding of many biological processes allowing pharmaceutical and biotechnology companies to determine for example new drug targets or to predict if particular drugs are applicable to all patients. Students will be introduced to the basic concepts behind Bioinformatics. Hands-on sessions will familiarize students with the details and use of the most commonly used online tools and resources.

Course Outline: This interdisciplinary course provides a hands-on approach to students in the topics of bioinformatics. Lectures and labs should cover sequence analysis, microarray expression analysis, Bayesian methods, control theory, scale-free networks, and biotechnology applications. Contents are designed for should include for those with a computational and/or engineering background, it will include current real-world examples, actual implementations, and engineering design issues. Where applicable, engineering issues from signal processing, network theory, machine learning, robotics and other domains will be expounded upon. The use of NCBI's Entrez, BLAST, PSI-BLAST, ClustalW, Pfam, PRINTS, BLOCKS, Prosite and the PDB.

Resources:

1. Jean-Michel Claverie, Cedric Notredame, Jean-Michel Claverie, " Bioinformatics for Dummies", John Wiley & Sons, Incorporated, 2007
2. Pierre Baldi, Søren Brunak (2001), "Bioinformatics: the machine learning approach", MIT Press.
3. Arthur M. Lesk (2002), "Introduction to Bioinformatics", Oxford University Press.
4. David R. Westhead, John Howard Parish, Richard M. Twyman (2002), "Bioinformatics", Published by BIOS.

Elective Supporting Courses

Course Name: **Computational Linear Algebra**

Course Structure: Lectures: 2 / Labs: 1 **Credit Hours:** 3

Prerequisites: Linear Algebra, Calculus

Objectives: Students will gain familiarity with and facility in the use of standard techniques for the numerical solution of a variety of problems in linear algebra, including solutions of linear systems, various matrix operations, evaluation of determinants and permanents, calculation of eigenvalues and determination of eigenvectors. Students will be introduced to various discrete transforms and apply some specific transforms to the solution of simple problems. In all cases, students will be introduced to possible sources of error and techniques for estimating the magnitude.

Course Outline: Background matrix algebra, measuring vectors, matrices, subspaces, and linear system sensitivity, numerical matrix algebra, Gaussian elimination, special linear systems, orthogonalization and least squares methods, the unsymmetrical eigenvalues problem, the symmetric eigenvalues problem, Lanczos methods, iterative methods for linear systems, functions of matrices. Introduction of discrete transforms, discrete Fourier and cosine transforms and simple applications. Error analysis and estimation for all techniques studied.

Sample labs and assignments:

- Implementation and testing of algorithms for typical linear algebra problems, including an analysis of errors.

Resources:

1. Golub, G., and C. Van Loan, *Matrix Computations (3/e)*, 1996
2. Sewell, G., *Computational Methods of Linear Algebra (2/e)*, 2005
3. Nievergelt, Y., *Wavelets Made Easy*, 1999.

Course Name: **Mathematical Tools for Software Engineering**

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: Linear Algebra, Calculus, Discrete Mathematics

Objectives: Students will gain familiarity with and facility in the use of a variety of mathematical concepts and tools with significant applications in software engineering, including mathematical models of machines and computations, various discrete and continuous optimization techniques, tools for the analysis of efficiency, formal methods for program verification

Course Outline:

Sample labs and assignments:

- Building a significant project using one or more well known middleware architectures.

Resources:

1. Graham, R. L., D. E. Knuth and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science (2/e)*, 1994
2. Greene, D. H., and D. E. Knuth, *Mathematics for the Analysis of Algorithms*, 1990.

Course Name: Operations Research

Course Structure: Lectures: 2 / Labs: 1 **Credit Hours:** 3

Prerequisites: Linear Algebra, Calculus, Discrete Mathematics, Probability and Statistics

Objectives: Students will become familiar with techniques of modeling real world problems. They will gain facility in working with a number of the most common models and modeling patterns. They will understand and be able to apply the notions of sensitivity analysis. They will be able to select appropriate deterministic or stochastic models in a wide variety of common situations.

Course Outline: Introduction to mathematical modeling. Linear program models, simplex method for solving LP models, sensitivity analysis, other solution techniques for LP models, specialized LP models (transport, assignment, etc.). Network based models, shortest path, min weight spanning tree, max flow, PERT/CPM. Decision models, dynamic programming, games theory. Probabilistic models, expected return models, Markov chains, stochastic processes, queueing models, stochastic inventory models.

Sample labs and assignments:

- Given a scenario, select and develop an appropriate model, solve it for the given parameters, and analyze the sensitivity of the solution to changes in the problem parameters.

Resources:

1. Hamdi A. Taha, *Operations Research: An Introduction (8/e)*, 2006
2. Hillier, F. S., and G. J. Lieberman, *Introduction to Operations Research (8/e)*, 2005

Course Name: Simulation and Modeling

Course Structure: Lectures: 2 / Labs: 3 **Credit Hours:** 3

Prerequisites: Probability and Statistics, Calculus

Objectives: This course emphasizes the development of modeling and simulation concepts and analysis skills necessary to design, program, implement, and use computers to solve complex systems/products analysis problems regarding software engineering discipline. The key emphasis is on problem formulation, model building, data analysis, solution techniques, and evaluation of alternative designs/processes in complex systems/products. Overview of modeling techniques and methods used in decision analysis, including Monte Carlo simulation and systems dynamics modeling are presented.

1. To apply modern software packages to conduct analysis of real world data.
2. To understand the technical underpinning of modern computer simulation software.
3. The ability to apply the appropriate analytical technique to a wide variety of real world problems and data sets.
4. To summarize and present the analysis results in a clear and coherent manner.

Course Outline: Introduction to Simulation and Modeling, Discrete-Event Simulation, Simulation of a Single-Server Queueing System, Alternative Approaches to Modeling and Simulations; Review of Basic Probability and

Statistics; Estimation of Means, Variances, and Correlations, Confidence Intervals and Hypothesis Tests for the Mean, The Laws of Large Numbers; Random number generators; Simulation of discrete, continuous probability distributions and empirical distributions; tests on simulated distributions, rejection method, simulation of multivariate distributions, correlations, and stochastic processes, simulation of models of arrival processes, Poisson Processes, Nonstationary Poisson Processes, Batch Arrivals, tests on generators, Markov- Chain Monte-Carlo simulations; Variance-Reduction Techniques.

Resources:

1. A.M. Law and W.D. Kelton, "Simulation Modeling and Analysis", McGraw Hill, 2000.
2. J. Banks, J.S. Carson and B.L. Nelson, "Discrete-event System Simulation", Prentice Hall International, 1994.
3. Mitrani, "Probabilistic Modeling", Cambridge University Press, 1998.
4. Sheldon M. Ross, "Simulation and Modeling", 2002.
5. Brian Ripley, "Stochastic Simulations".

Course Name: Ethics and Professional Practices for Computing Professionals

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives: A Computing graduate as professional has some responsibilities with respect to the society. This course develops student understanding about historical, social, economic, ethical, and professional issues related to the discipline of Computing. It identifies key sources for information and opinion about professionalism and ethics. Students analyze, evaluate, and assess ethical and professional computing case studies.

Course Outline: Introduction, Computing Ethics, Philosophy of Ethics, Ethics and the Internet.
Intellectual Copy Right, Accountability and Auditing, Social Application of Ethics.

Resources:

1. Deborah G. Johnson, "Computer Ethics", Pearson Education (2001) 3rd edition.
2. *Professional Issues in Software Engineering*, M.F. Bott et. al.

Curriculum for MS Software Engineering — MS (SE)

Eligibility

1. BS (SE/CS) 4 years degree programme,
OR
2. Computer Science conversion course two years degree programme referred to as MCS or M.Sc. (Computer Science),
OR
3. BCS 3-year programme degree applicants may be provisionally admitted in the MS (SE) programme. Candidates will be required to take additional courses to complete credit hour requirement of min. 130 before being formally enrolled in the MS (SE) programme.

Under eligibility criteria 1-3 the university/department may recommend additional deficiency courses, from the BS (SE) curriculum, considering the deficiency of the candidates.

OR

4. 16-years education science/engineering degrees.

Under eligibility criterion 4 candidates will be required to complete the deficiency coursework prior to the MS (SE) coursework to ensure the pre-requisite competency in SE.

The deficiency coursework will be determined on the basis of the core SE courses of the BS (SE) degree.

Duration

- 4 semesters
- 30-36 credit hours from graduate Software Engineering courses including thesis

Degree Requirements

In order to obtain MS (SE) degree a student must pass a minimum of:

- i) Four (4) courses (12 credit hours) from the core courses
AND
- ii) Four (4) courses of 12 credit hours graduate elective courses of which two graduate courses may be taken from other areas.
AND
- iii) Satisfactorily complete a *Research Project Thesis* of 9 credit hours.

Core Courses

Following three courses are the core

S. No.	Code	Course Title	Cr. Hrs.	Semester
1	SE	Requirement Engineering	3	1-2
2	SE	Software System Architecture	3	1-2
3	SE	Software System Quality	3	1-2

Elective Courses

- Candidate has to select a minimum of Two (2) courses from the following list of SE electives. Other electives may be taken from allied areas to support the research work.

Graduate Level SE courses (Institution may add courses to the list of Electives.)

Elective Courses			
1	Software Engineering Management	10	Formal Methods in Software Engineering
2	Software Risk Management	11	Software Engineering Ontologies
3	Software Measurement and Metrics	12	Semantic based Software Development
4	Global Software Engineering	13	Semantic web enabled software engineering
5	Software Configuration Management	14	Model Driven Software Development
6	Knowledge Based software Engineering	15	Machine Learning Applications in Software Engineering
7	Software Dependability	16	Software Process Engineering
8	Software Costing and Estimation	17	Software Case tools and Applications
9	Business Process Re-engineering	18	Web Engineering

Sample Scheme of Study for MS (SE)
2-year Programme (4 Semesters)

(30 Credit Hours)

Semester 1

S.No.	Code	Course Title	Cr. Hrs.
1	SE	Requirement Engineering	3
2	SE	Software System Architecture	3
3	SE	Elective-I	3

Total 9 Credit Hrs.

Semester 2

S.No.	Code	Course Title	Cr. Hrs.
1	SE	Software System Quality	3
2	SE	Elective II	3
3	SE	Elective-III	3

Total 9 Credit Hrs.

Semester 3

S.No.	Code	Course Title	Cr. Hrs.
1	SE	Elective-IV	3
2	SE	Elective-V	3
3	SE	Thesis-I	3

Total 9 Credit Hrs.

Semester 4

S.No.	Code	Course Title	Cr. Hrs.
1	SE	Thesis-II	6

Total 33 Credit Hrs.

MS (SE) – Core Courses

Course Name: Requirement Engineering

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives: To understand Issues in Requirements Engineering, to understand and apply Requirements Engineering Process, to understand and use Requirements Elicitation and Specification, to understand and use Formal Techniques, to understand modeling and analysis of Non-Functional Requirements.

Course Outline: Definition of requirements engineering and role in system development, Fundamental concepts and activities of requirements engineering, Information elicitation techniques, Modeling scenarios

Fundamentals of goal-oriented requirements engineering, Modeling behavioral goals, Modeling quality goals, Goal modeling heuristics, Object modeling for requirements engineering, Object modeling notations, Object modeling heuristics, Identifying objects from goals, Modeling use cases and state machines, Deriving operational requirements from goals, Requirements Specification, Requirements verification and validation

Management of inconsistency and conflict, requirements engineering risks, the role of quality goals in the requirements selection process, Techniques for requirements evaluation, selection and prioritization; Requirements management; Requirements traceability and impact analysis.

Resources:

1. Hull, Jackson, and Dick, Requirements Engineering, 2004, Springer
2. Karl E. Wiegers, Software Requirements, 2nd Edition, 2003, Microsoft Press.
3. Loucopoulos and Karakostas, System Requirements Engineering, McGraw-Hill.
4. Kotonya and Sommerville, Requirements Engineering: Processes and Techniques, John Wiley Sons.

Course Name: Software System Architecture

Course Structure: Lectures: 3 **Credit Hours:** 3

Prerequisites: None

Objectives: To develop an understanding of the relationships between system qualities and software architectures, software architectural styles and their relationship to system qualities, software architecture evaluation, attribute-driven design, software architecture documentation, architectural reuse.

Course Outline: Definition and overview of software architecture, the architecture business cycle, Understanding and achieving quality attributes, Attribute-driven design, Documenting software architecture, Evaluating software architecture, Architecture reuse

Life-cycle view of architecture design and analysis methods, The QAW, a method for eliciting critical quality attributes, such as availability, performance, security, interoperability, and modifiability, Architecture Driven Design, Evaluating a software architecture (ATAM, CBAM, ARID), Principles of sound documentation, View types, styles, and views; Advanced concepts such as refinement, context

diagrams, variability, software interfaces, and how to document interfaces; Documenting the behavior of software elements and software systems; Choosing relevant views; Building a documentation package.

Resources:

1. Taylor, Medvidovic, and Dashofy, *Software Architecture: Foundations, Theory, and Practice*, 2009
2. Anthony J. Lattanze, *Architecting Software Intensive Systems: A Practitioners Guide*, 2008, Auerbach Publications
3. Bass, Clements, and Kazman, *Software Architecture in Practice (2nd Edition)*, 2003, Addison-Wesley Professional
4. Clements, Kazman, and Klein, *Evaluating Software Architectures: Methods and Case Studies*, 2001, Addison-Wesley Professional.

Course Name: Software System Quality

Course Structure: Lectures: 3

Credit Hours: 3

Prerequisites: None

Objectives: The objective of this course is to study in detail the issues involved in software quality engineering. The course focuses on current practice, research and trends in Quality: how to assure it and verify it, and the need for a culture of quality. Avoidance of errors and other quality problems. Inspections and reviews. Testing, verification and validation techniques. Process assurance vs. Product assurance. Quality process standards. Product and process assurance. Problem analysis and reporting. Statistical approaches to quality control. Economics of testing, verification and validation activities, and software quality improvement through systematic test planning, design and executions, problem reporting and resolutions, **and** test documentation. Establishing software quality goals and improvement measurement.

Course Outline:

What Is Software Quality: Quality Assurance, Quality Engineering

SOFTWARE TESTING: Testing: Concepts, Issues, and Techniques, Test Activities, Management, and Automation, Coverage and Usage Testing Based on Checklists and Partitions, Input Domain Partitioning and Boundary Testing, Coverage and Usage Testing Based on Finite-State Machines and Markov Chains, Control Flow, Data Dependency, and Interaction Testing, Testing Techniques: Adaptation, Specialization, and Integration.

QUALITY ASSURANCE BEYOND TESTING: Defect Prevention and Process Improvement, Software Inspection, Formal Verification, Fault Tolerance and Failure Containment, Comparing Quality Assurance Techniques and Activities.

QUANTIFIABLE QUALITY IMPROVEMENT: Feedback Loop and Activities for Quantifiable Quality Improvement, Quality Models and Measurements, Defect Classification and Analysis.

Risk Identification for Quantifiable Quality Improvement, Software Reliability Engineering.

Sample labs and assignments

- Use of automated testing tools

- Testing of a wide variety of software
- Application of a wide variety of testing techniques
- Inspecting of software in teams; comparison and analysis of results

Resources:

1. Jeff Tian (2005), *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement*,” Wiley-IEEE Computer Society Pre
2. Boris Beizer, *Software Testing Techniques (second edition)*.
3. *Current research publications and literature and URLs where such courses are being offered.*

Elective Courses

Course Name: Software Engineering Laboratory

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours: 3**

Prerequisites:

Objectives:

Course Outline: This course is designed to help the student develop the capability in specifying, designing, and implementing real-life software applications following software development methodologies. Each student/team is expected to select an area of greatest interest and implement a related general interest software application. Software application areas covered include. Multimedia, Client server, Internet/network computing, database systems, objects oriented modeling, software quality documentation, and testing and project management. The evaluation will be done in a similar manner as outlined for software project. The results for the continuous assessment (30%) will be submitted by the course supervisor. Final Project workshop results will be prepared by the supervisor based on the final evaluation (70%) by a panel of IT experts, Client Representatives and Internal Supervisor. All students/groups /projects will be examined by the panel.

Course Name: **Research Study**

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours: 3**

Prerequisites:

Objectives:

Introduction: The aim of the course is “to provide the students with an ability to undertake postgraduate level research and an appreciation of relevant ethical, professional and legal issues”. Essentially the ethical, professional and legal issues provide a research domain but the overall purpose is to ensure that the students gain research skills that will support them in the rest of their courses, in their future careers, and that they will undertake their research activities in an ethical and professional manner.

Abilities:

1. Can carry out research investigations using information repositories.
2. Can effectively report the results of research activities
3. Can develop and deliver presentations to disseminate research findings

Course Outline:

Introduction to the course:

- International Ethical, Professional and Legal Issues in Computing
- Introduction to the Concepts of Research-1: Definitions, Quantitative and Qualitative Approaches
- Introduction to the Concepts of Research-2: Process, Forming Hypotheses, Originality, Critical analysis
- Thinking about methods
- Reading for research
- Data Collection and Information Gathering
- Information Gathering: Literature Surveys
- Data Analysis
- Proposals for Research Projects and Research Papers
- Information Gathering: Surveys and Questionnaires
- Presentation of Information: Writing Academic Papers-1-
- Content and Referencing

The students have to perform meta analyses of 25-30 research papers selected in current research topics in International Journals. Topic and papers will be selected with approval from the instructor. Conference papers are not allowed for review. Students have to read all such papers and prepare the analysis related to model, methods, findings and come up with what has been done related to selected area of research and research gaps if any are explicitly identified with future work.

Resources:

1. How to Research, 2nd Ed, by Loraine Blaxter, C. Hughes, M. Tight

Course Name: Software Case Tools & Applications

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites:

Objectives:

Course Outline: The students will be appraised of; Case tools & techniques, CASE in software development process, Traditional CASE methodologies, Emerging CASE methodologies, OO Design, Specific CASE tools, specialized design tools, Managing CASE methodologies.

As part of course, students will be assigned a real life problem for development through CASE tools

Resources: Selected software case tool documentation

Course Name: Software Engineering Ontologies

Course Structure: Lectures: 3 Credit Hours: 3

Prerequisites: None

Objectives: The objective of this course is to study in detail the Ontologies available for software development and highlights their strengths and weaknesses in achieving the goals for which the Ontologies have been developed. The course starts from the introduction to Ontologies and latest

languages used to describe / document Ontologies. Use of Ontologies and its significance in development of software systems will be covered with the help of some real life examples. Then a detailed study and comparison of different Ontologies available for each phase in the software engineering development life cycle will be done.

Course Outline:

1. Ontology Engineering: Principles, Methods, Tools, and Languages
2. Using Ontologies in Software Engineering
3. Development of Ontologies for SWEBOK (Software Engineering Body of Knowledge): Issues and Techniques
4. Some Ontologies for Software Development: Ontologies for Requirements, Design, Maintenance, Measurements,
5. Use of Ontologies in Domain Oriented Software Development Environments
6. Comparative Study of Semantics Coverage in Ontologies as per SWEBOK
7. Alignment of Different Available Ontologies.

There will be a lot of case studies in this course as assignments.

Resources:

1. Extensive Use of Online Available Latest Resources
2. Online Available Ontologies (search by using Swoogle)
3. Ontologies for Software Engineering and Software Technology
by [Coral Calero](#) (Editor), [Francisco Ruiz](#) (Editor), [Mario Piattini](#) (Editor)
Publisher: Springer; 1 edition (October 19, 2006) Language: English ISBN-10:
3540345175 ISBN-13: 978-3540345176

National Curriculum Revision Committee (NCRC): Information Technology (IT)-2009

A three-day final meeting of National Curriculum Revision Committee was held from April 21-23, 2009 at Higher Education Commission (HEC), Islamabad. The purpose of this meeting was to finalize the draft curricula for undergraduate as well as graduate students of Information Technology. The lengthy discussions held throughout the period finally led us to design the curricula for BS, MS/MPhil degree programs. The following experts participated in the meeting:

- 1. Professor Dr Farhana Shah** **Convener**
Director
Institute of Information Technology
Quaid-i-Azam University
Islamabad
- 2. Professor Dr Imdad Ali Ismaili** **Secretary**
Director
Institute of Information Technology
University of Sindh
Jamshoro
- 3. Dr Naveed Ikram** **Member**
Associate Professor
Department of Computer Science
International Islamic University
Islamabad
- 4. Professor Dr Muhammad Sher** **Member**
Chairman
Department of Computer Science
International Islamic University
Islamabad
- 5. Dr Muhammad Yousaf** **Member**
Associate Professor
Department of Computer Science & Engineering
Bahria University
Islamabad
- 6. Dr. Muhammad Ali** **Member**
Assistant Professor
Department of Information Technology & Computer Science
Institute of Management Sciences
Peshawar
- 7. Professor Dr. Nazir A. Sangi** **Member**
Chairman
Department of Computer Science
Allama Iqbal Open University

Islamabad

- 8. Mr. Muhammad Nadeem Khokhar** **Member**
Assistant Professor and Coordinator
Computer Science Department
Shaheed Zulfikar Ali Bhutto Institute of Science & Technology
Islamabad
- 9. Dr. Sohail Asghar** **Member**
Assistant Professor and Head of R&D
Department of Computer Science
Shaheed Zulfikar Ali Bhutto Institute of Science & Technology
Islamabad
- 10. Professor Dr Iftikhar Hussain Shah** **Member**
Professor
Department of Computer Science & Information Technology
Forman Christian College
Lahore
- 11. Professor Dr Madad Ali Shah** **Member**
Professor
Information Technology
IBA Sukkur Airport Road
Sukkur
- 12. Professor Dr Jerald Allan Kabell** **Member**
Chairperson
Department of Computer Science & Information Technology,
Forman Christian College
Lahore
- 13. Dr Sharifullah Khan** **Member**
Associate Professor
School of Electrical Engineering & Computer Sciences (SEECS)
National University of Sciences and Technology
Rawalpindi
- 14. Professor Dr Aqil Burney** **Member**
Chairman
Department of Computer Science and Information Technology
University of Karachi
Karachi.
- 15. Dr Shafay Shamail** **Member**
Associate Professor
Department of Computer Science
LUMS
Lahore

- 16. Professor Dr Zubair A Shaikh** **Member**
Representative, NCEAC
FAST National University of Computer & Emerging Sciences
Karachi
- 17. Professor Dr Abdul Qadir** **Member**
Dean
Faculty of Engineering and Sciences
Muhammad Ali Jinnah University (MAJU)
Islamabad
- 18. Professor Dr. Syed Mansoor Sarwar** **Member**
Principal
PU College of Information Technology (PUCIT)
Punjab University
Lahore
- 19. Professor Dr. Abdul Hussain Shah Bukhari** **Member**
Dean
Faculty of Information and Communication Technology
Balochistan University of Information Technology, Engineering, and
Management Sciences, Quetta

First Meeting of National Curricula Revision Committee in the Field of Information Technology

The meeting started with recitation of the holy Quran. Dr. Riaz ul Haq, Member (Acad) presided over the meeting. He welcomed the participants and highlighted the need for reviewing the existing curriculum.

The members of the National Curricula Revision Committee (NCRC) (the Committee) unanimously nominated and elected Dr Farhana Shah as Convener and Dr Naveed Ikram as Secretary of the Committee.

The Convener declared the floor open for discussion after brief introductory remarks and explaining rules of the game. The participants liked to begin the revision of the existing curriculum in light of:

- a) Changes already recommended by Computer Science Committee especially bringing in of the common section of Computing Part for undergraduates
- b) Revised modifications recommended by the international community (e.g. ACM/IEEE) on previous curricula suggested in IT
- c) The feedback and innovative ideas of members of the committee based on their experiences and diverse backgrounds

Revision of Goals for the Program of BS in Information Technology

The participants of the committee preferred to discuss the product of the program by having a vision and setting the goals first. A sub committee was assigned the task of researching the effective goals for the next four years at least. The work was presented before the committee for deliberation. Following are the recommendations by the committee as a result of combined consensus.

The aim of the undergraduate program of IT is to provide students with skills and knowledge that enable them to take on appropriate professional positions in IT and grow into leading roles. The goals are to produce, in coordination with organizational management, IT graduates who can:

- a) identify needs and possibilities of the organization which may be met by appropriate use of IT resources, including hardware, software and communication technologies;
- b) plan, select, integrate, deploy, manage and support the required IT resources;
- c) communicate with a range of audiences and participate effectively as part of teams;
- d) analyze the local and global impact of computing and understand professional, ethical, legal, security and social issues, and their responsibilities as IT professionals; and
- e) help in improving the quality for higher education programs.

Review of Recommendations Made by NCRC for Computer Science (2008) Regarding “Computing” Section

The NCRC for Information Technology agreed to disagree upon the Computing Part with the following observations and recommendations:

- a) “Introduction to Computing” be renamed to “Introduction to Information and Communication Technologies (ICT)” and be moved from “Computing - Core Courses” to “Computing - General Education”.
- b) The course “Human Computer Interaction” be added to the “Computing–Core Courses.”
- c) The titles of the courses “Digital Logic and Computer Architecture” and “Database Systems” in Computing – Core Courses be changed to “Digital Logic Design” and “Introduction to Database Systems” respectively.
- d) A course of “Basic Electronics” be introduced in place of “Physics (Electromagnetism)” in “Computing - Supporting Sciences” part.
- e) The course of “Professional Practices” in “Computing – General Education” should cover Social, Ethical, Legal and Professional issues.
- f) The Islamic and Pakistan Studies course should be divided in to two courses of 2 credit hours each.
- g) The NCRC for Information Technology (IT) did not agree with the recommendation(s) coming from the NCRC for Computer Science (CS) that the course “Discrete Structures” should be moved from the Computing - Core Courses” to “CS Required Supporting Courses”. However, it was recommended by the NCRC for IT that this course should remain within “Computing–Core Courses”.

Revision of BS Program in Information Technology

The task was divided into subtasks. Three sub-committees were constituted to pay special attention to details, revise three subsets of courses and work thoroughly on their contents. The three subsets of courses revolved around Technology, Strategy, and Management respectively. The lay out of courses together with the contents and up to date books were brought to the main committee for further discussion. A consensus was built on recommendations as given below:

- a) Six courses worth 18 credit hours should be considered Core area for the curriculum of BS in Information Technology as follows:
 - a. Fundamentals of Information Technology
 - b. Web Systems and Technologies
 - c. Multimedia Systems and Design
 - d. Systems and Network Administration
 - e. Network Security
 - f. System Integration and Architecture
- b) The course of “Principles of Management” should be replaced by “Technology Management” in the Required Supporting area defined for the curriculum.
- c) The Elective courses with respect to Information Technology and General areas were suggested along with Fields of Concentration as follows:
 - a. Communication Systems Design
 - b. Information Security
 - c. Mobile and Pervasive Computing
 - d. Web Services
 - e. Web Site design and Usability
 - f. Knowledge-Based Systems
 - g. Data Warehousing
 - h. Data Mining
 - i. Web Technologies and e-Systems
 - j. Network Systems
 - k. Knowledge Management

However the list is suggestive not exhaustive, universities may offer other courses.

Structure of BS in Information Technology

#	Category		Credit Hours
1	Computing Courses		68
	Core Courses	37	
	Supporting Areas	13	
	General Education	18	
2	Information Technology Courses		48
	IT Core Courses	18	
	IT Electives Courses	21	
	IT Supporting Courses	9	
3	University Electives		18

Total Credit Hours	134
--------------------	-----

Computing — Core Courses (37 Credits Hours)

Required Computing Courses					
#	Code	Prereq	Course Title	Credit hours	Proposed Semester
1	CS	-	Programming Fundamentals	4 (3-1)	1
2	CS	1	Object Oriented Paradigm	3 (2-1)	2
3	CS	-	Discrete Structures	3 (3-0)	2
4	CS	2	Data Structure and Algorithms	3 (2-1)	3
5	CS	3	Digital Logic Design	3 (2-1)	3
6	CS	4	Operating Systems	3 (2-1)	4
7	CS	4	Introduction to Database Systems	3 (2-1)	4
8	CS	4	Introduction to Software Development	3 (3-0)	5
9	CS	6	Computer Communications and Networks	3 (2-1)	6
10	CS		Human Computer Interaction	3 (3-0)	
11	CS	-	IT Capstone	6 (0-18)	7, 8
(37/134)					

Computing — Supporting Sciences (12 Credits Hours)

Required Supporting Courses					
#	Code	Prereq	Course Title	Credit hours	Proposed Semester
12	MT	-	Calculus and Analytical Geometry	3 (3-0)	1
13	MT	-	Probability and Statistics	4 (4-0)	2
14	MT	-	Linear Algebra	3 (3-0)	4
15	EE	-	Basic Electronics	3 (3-0)	3
(13/134)					

Computing — General Education (18 Credits Hours)

Required General Education Courses					
#	Code	Prereq	Course Title	Credit hours	Proposed Semester
1	EG	-	English-I (Functional English)	3 (3-0)	1
2	EG	-	English-II (Technical and Report Writing)	3 (3-0)	2
3	EG	-	English-III (Communication Skills)	3 (3-0)	3
4	PK	-	Islamic and Pakistan Studies	3 (3-0)	1

5	IT	-	Introduction to Information and Communication Technology	3(2-1)	1
6	SS	-	Professional Practices	3 (3-0)	8
			(18/134)		

IT — Core Courses (18 Credits Hours)

Required IT Core Courses					
#	Code	Prereq	Course Title	Credit hours	Proposed Semester
1			Fundamentals of Information Technology	3 (3-0)	
2			Web Systems and Technologies	3(2-1)	
3			Multimedia Systems and Design	3(2-1)	
4			Systems and Network Administration	3(3-0)	
5			Network Security	3(3-0)	
6			System Integration and Architecture	3(3-0)	
			(18/134)		

IT — Supporting Sciences (9 Credits Hours)

Required Supporting Courses					
#	Code	Prereq	Course Title	Credit hours	Proposed Semester
			Technology Management	3 (3-0)	
			Organizational Behaviour	3 (3-0)	
			Information Systems	3 (3-0)	
			(9/134)		

IT — IT Electives (21 Credits Hours)

Following is a suggestive list of the elective courses. Universities may offer other courses.

IT Electives					
#	Code	Prereq	Course Title	Credit hours	Proposed Semester
1			Communication Systems Design	3 (3-0)	
2			Information Security	3 (3-0)	
3			Communication Technologies	3 (3-0)	
4			Mobile and Pervasive Computing	3 (3-0)	
5			Web Services	3 (3-0)	
6			Web Site Design and Usability	3 (3-0)	
7			Knowledge-Based Systems	3 (3-0)	

8			Database Management	3 (3-0)	
9			Data Warehousing	3 (3-0)	
10			Information Retrieval	3 (3-0)	

Scheme of Study for BS (IT)

4-Year Program (8 Semesters)

(130 Credit Hours)

Semester-wise 4-Year Plan

Semester 1	Cr. Hrs.	Semester 2	Cr. Hrs.
Introduction to ICT Programming Fundamentals	2+1	Discrete Structures Object Oriented Programming	3+0
Calculus and Analytical Geometry	3+1	Fundamentals of IT	2+1
Basic Electronics	3+0	University Elective I	3+0
English-I (Functional English)	2+1	English-II (Communication Skills)	3+0
	3+0	Pakistan Studies	2+0
	16		17
Semester 3	Cr. Hrs.	Semester 4	Cr. Hrs.
Digital Logic Design	2+1	Operating Systems	2+1
Data Structures and Algorithms	2+1	Introduction to Database Systems	2+1
Linear Algebra	3+0	Organizational Behaviour	3+0
English-III (Technical and Report Writing)	3+0	Probability and Statistics	3+0
Islamic Studies/Ethics	2+0	Computer Communication and Networks	2+1
University Elective II	3+0	University Elective III	3+0
	17		18
Semester 5	Cr. Hrs.	Semester 6	Cr. Hrs.
Web Systems and Technologies	2+1	University Elective V	3+0
University Elective IV	3+0	Systems and Network Administration	2+1
Introduction to Software Development	3+0	IT Elective II	3+0
IT Elective I	3+0	University Elective VI	3+0
Multimedia Systems and Design	2+1	IT Elective –III	3+0
Information Systems	3	Human Computer Interaction	2+1
	18		18

Semester 7	Cr. Hrs.	Semester 8	Cr. Hrs.
IT Capstone Part I (continued)	*	IT Capstone Part II	6
Technology Management	3	IT Elective VI	3
IT Elective IV	3	Professional Practices	3
Network Security	3	IT Elective VII	3
System Integration and Architecture	3		
IT Elective V	3		
	18		12

MS Program in Information Technology

The curriculum for the Master's program was thrashed out with diverse perspectives. Everybody agreed upon defining tracks consistently and suggesting courses accordingly. Finally, after incorporating the approved changes the structure of MS in IT with its complete design and details emerged as follows:

Structure of MS in Information Technology

Category or Area	Credit Hours
Core	12
Elective	12
Thesis/Project/Course work	6
Total Credit Hours	30

Core Area

S No	Course Title	Credit Hours
1	Advanced Database Management Systems	3
2	Telecom Management	3
3	Information Security and Assurance	3
4	Information Technology Infrastructure	3

Elective Area

The committee argued at length the elective courses and recommended the following courses as suggestive list. Universities may add more courses on similar lines.

S No	Course Title	Credit Hours
1	Economics of Technology	3
2	IT Planning and Evaluation	3
3	IT Services Management	3
4	IT Project Management	3
5	E-Biz	3
6	IT Audit and Assessment	3
7	IT Policy, Laws, and Practice	3
8	IT Disaster Management	3
9	Distributed Databases	3
10	Data Mining	3
11	Advanced Topics in Databases	3
12	Information Technology Architecture	3

Thesis/Project/Course work

The committee, after long discussion, recommended that university should be given option for selecting thesis, project work, or course work. A minimum of 6 credit hours for thesis/project work/course work are recommended.

COURSE CONTENTS

For BS in IT

Course Name: Introduction to Database Systems

Course Structure: Lectures: 2/Labs: 3 **Credit Hours:** 3

Prerequisites: Data Structures and Algorithms

Objectives:

The course aims to introduce basic database concepts, different data models, data storage and retrieval techniques and database design techniques. The course primarily focuses on relational data model and DBMS

Course Outline:

Basic database concepts; Logical database Modelling and design: Entity Relationship diagram (ERD), Enhanced ERD Relational data model: mapping ERD to relational model, Functional dependencies and Normalization: 1st -3rd Normal Form and BCNF, Relational Algebra; Structured Query language (SQL); Fundamental knowledge about Transaction processing, concurrency control recovery techniques and query optimization concepts.

Reference Material:

1. C. J. Date, Database Systems, Addison Wesley Pub. Co.
2. R. Elmasri and S. Navathe. Fundamentals of Database Systems, Benjamin/Cummings.
3. Abraham Silberschatz, Henry F. Korth S. Sudarshan. "Database System Concepts".
4. T.Connolly and C.Begg . "Database Systems, a Practical Approach to Design, Implementation and Management", Pearson education,.

Computing: General Education

Course Name: Introduction to Information and Communication Technologies

Course Structure: Lectures: 2 / Labs: 3 **Credit Hours:** 3

Prerequisites: None (first semester course)

Objectives:

This course focuses on a breadth-first coverage of the use of computing and communication technologies to solve real life problems; including computing environments, general application software like word processing, visual presentation applications, tabular data manipulation, DBMS, WWW, Email management systems, Virus, Anti-Virus and Spam Protection; Introduction to the basic computing hardware (main building blocks), operating systems, data networks; software engineering and communication technology along with social and ethical issues. An introduction of the program of study in computing for which this course is being taught (CS, IT, SE etc.). The course attempts to provide every student a set of productivity tools that they will be able to use for the rest of their lives.

Course Outline:

Number Systems, Binary numbers, Boolean logic, History computer system, basic machine organization, Von Neumann Architecture, Algorithm definition, design, and implementation, Programming paradigms and languages, Graphical programming, Overview of Software Engineering and Information Communication Technology, Operating system, Compiler, DBMS, Computer networks and internet, WWW, web mail applications, Computer graphics, AI, Viruses and Anti-Viruses, Use of office productivity tools, such as word processors, spreadsheets, presentation applications, etc., Social, Ethical, Professional and Legal Issues, and overview of the complete program of studies in computing and its structure.

Suggested Text Book:

1. Introduction to Computers by Peter Norton, 6th Edition, McGraw-Hill SiE, ISBN 0-07-059374-4.

Reference Material:

1. Computers: Information Technology in Perspective, 9/e by Larry Long and Nancy Long, Prentice Hall, 2002/ISBN: 0130929891.
2. An Invitation to Computer Science, Schneider and Gersting, Brooks/Cole Thomson Learning, 2000.
3. Information System Today by Leonard Jessup, Joseph Valacich.
4. Computers Today by Suresh K. Basandra.
5. Computer Science: An overview of Computer Science, Sherer.

BS IT Core Courses**Course Name: Fundamentals of Information Technology****Course Structure:** Lectures: 3**Credit Hours:** 3**Prerequisites:** Introduction to Computing (recommended)**Objectives:**

To introduce students to the scope of the field of Information Technology, to give them a basic understanding of information, its organization, transmission, storage, retrieval and presentation, and to explore some of the computer based technologies used for these purposes.

Course Outline:

Introduction to the academic discipline of IT as well as the general meaning of IT as per objectives given in the start of this program. Definitions of information, information technology as the use of computer based technology to organize, store, retrieve, transmit and present information, sender/receiver/channel model for information transfer. Information organization via databases, data modeling, and information management systems. Basic network ideas and models. Differences in human and machine processing of information, information transfer at the human/machine interface, modalities for information presentation, advantages and disadvantages of various presentation media. Challenging issues for today's information and communication technologies, issues in organizational need assessment and management of large scale information systems, along with social, legal and ethical issues related with each topic.

Suggested Text Book:

1. Cyganski, David, John A. Orr and Richard F. Vaz, Information Technology Inside and Outside, Pearson Education (LPE), 2001
2. Information Technology: Principles, Practices, and Opportunities (3rd Edition) (Hardcover), by [James A. Senn](#) (Author), Prentice Hall; 3 edition (December 1, 2003), ISBN-10: 0131436260

Reference Material:

1. Cyganski, David, John A. Orr and Richard F. Vaz, Information Technology Inside and Outside, Pearson Education (LPE), 2001
2. Ray, Ajoy Kumar and Tinku Acharya, Information Technology: Principles and Applications, Prentice-Hall India, 2004
3. Information Technology: Principles, Practices, and Opportunities (3rd Edition) (Hardcover), by [James A. Senn](#) (Author), Prentice Hall; 3 edition (December 1, 2003), ISBN-10: 0131436260
4. Introduction to Information Technology (Hardcover), by [Efraim Turban](#) (Author), [Rex Kelly Rainer](#) (Author), [Richard E. Potter](#) (Author), Hardcover: 592 pages, Publisher: Wiley; 2 edition (July 12, 2002), ISBN-10: 0471073806

Course Name: Web Systems and Technologies

Course Structure: Lectures: 3 **Credit Hours:** 3

Prerequisites: Fundamentals of Information Technology (required)

Objectives:

This course will extend the WWW Technologies and Web Based Applications architecture, development, deployment and management concepts studied in the course of Fundamentals of Information Technology. The instructor is expected to cover an in-depth treatment of the web technology and applications related topics including web standards, protocols, web applications architecture, web services, search engine architectures, content management, web2, and semantic web, to explore some of the technologies used for display, data access and processing, and to give the students practice in integrating these to produce a functional web-based system.

Course Outline:

In-depth study of World Wide Web architectures, protocols and standards (HTTP, HTML, XHTML, CGI, XML, WML, cHTML, etc.), Web Technologies and Tools (such as scripting tools) for web application development and deployment (web servers, application servers, etc.), Web Based Applications including search engines and content management, management of large scale web based information systems, Web Services, Web2, Semantic Web, and Web3, principles of web site design, practical exercise in web site development.

Suggested Text Books:

1. Nuckles, Craig, Web Applications: Concepts and Real World Design, Wiley 2006
2. Programming the World Wide Web (4th Edition) (Paperback), by [Robert W. Sebesta](#) (Author), Paperback: 752 pages, Publisher: Addison Wesley; 4th edition (August 17, 2007), ISBN-10: 0321489691

Reference Material:

1. Gosselin, Dan, et. al., The Web Warrior Guide to Web Design Technologies, Cengage Learning, 2003
2. Zak, Diane, et. al., The Web Warrior Guide to Web Programming, Cengage Learning, 2003
3. Leasure, T., Bob Leasure and James Leasure, The Web Warrior Guide to Web Database Technologies, Cengage Learning, 2003
4. Morrison, Mike and Joline Morrison, Database Driven Websites, 2/e, Cengage Learning, 2002
5. Web Wizard series for various technologies, Addison-Wesley
6. Jackson, J. C., Web Technologies: A Computer Science Perspective, Pearson (LPE), 2008
7. [Web Application Architecture: Principles, Protocols and Practices](#) by Leon Shklar and Richard Rosen (Paperback - Oct 31, 2008), Paperback: 420 pages, Publisher: Wiley; 2 edition (October 31, 2008), ISBN-10: 047051860X
8. [Web Engineering: The Discipline of Systematic Development of Web Applications](#) by Gerti Kappel, Birgit Prýýll, Siegfried Reich, and Werner Retschitzegger (Paperback - Jul 5, 2006)

Course Name: Multimedia Systems and Design

Course Structure: Lectures: 2, Lab: 3 **Credit Hours:** 3

Prerequisites: Fundamentals of Information Technology (required)

Objectives:

To introduce students to the complete process of multimedia system specification, design, testing, and prototyping, including the tools and techniques for integrating multimedia content (text, graphics, images, sound, animation, motion video and virtual reality) into a product, to present design principles and techniques to maximize the effectiveness of such products, and to give the students practice in the production using a variety of media and tools. Introduction to multimedia systems, multimedia applications and development tools.

Course Outline:

Introduction to multimedia systems, software, hardware, various equipment, video and audio capture, annotation, storage and playback techniques, multimedia software development tools, multimedia applications, step-by-step procedure in developing multimedia systems: (specification, design, testing, and prototyping), multimedia standards, Student projects - developing multimedia systems in the laboratory.

Suggested Text Books:

1. [Multimedia: Making it Work, Seventh Edition](#) by Tay Vaughan (Paperback - Dec 20, 2006)
2. Shuman, James, Multimedia Concepts, Enhanced Edition, Cengage Learning, 2002
3. Lake, Susan and Karen Bean, Digital Multimedia: The Business of Technology, Cengage Learning, 2007

Reference Material:

1. Z. M. Li; M. S. Drew: Fundamentals of Multimedia. Prentice Hall 2004, ISBN: 0-13-127256-X
2. N. Chapman; J. Chapman: Digital Multimedia. (2nd ed.), Wiley 2004, ISBN: 0-470-85890-7
3. Villalobos, Ray, Exploring Multimedia for Designers, Cengage Learning, 2007.

Course Name: System and Network Administration

Course Structure: Lectures: 2/Labs: 1 **Credit Hours:** 3 **Semester:** 5

Suggested Prerequisites: Computer Communication and Networks, Operating Systems

Objectives:

This course will give an overview of systems and network administration based on both Windows and Linux environments. The objective are common system administration tasks and practices and how to implement and maintain standard services like email, file sharing, DNS and similar. The course is primarily dealing with the Linux and Windows operating systems and especially with Linux-based servers and Window-based clients, but some information about the most fundamental differences between various Linux systems will be provided. In labs focus is on how to install, setup and maintain Linux server machine and to perform various system administration and security related tasks on those machines.

Course Outline:

Brief introduction to the Networks, Homogenous and Heterogeneous networks, Issues involved in the setup of Heterogeneous networks, File systems, Configuration issues, Fundamentals of Linux user interface, Installation and administration of heterogeneous networks using Windows and Linux platforms. System installation, booting and halting the system, file systems and directory permission structures, print and disk quotas, device configuration and management, user account administration, security, client administration, disk maintenance, remote access, remote administration, the use of schedulers, the use of advanced scripting to ease system administration tasks, configuration management, template implementation and cross directory implementation.

Suggested Textbooks:

Reference Material:

1. Practice of System and Network Administration, the 2nd Edition by Thomas A, Limoncelli, Hogan, 2005.
2. Windows Administration Latest Edition, Microsoft Press
3. Linux Administration Guide Latest Edition

Course Name: Network Security

Course Structure: Lectures: 3/Labs: 0 **Credit Hours:** 3 **Semester:** 7

Prerequisites: Computer Communication and Network

Course Outline:

Principles and Practices of network security, security threats and methods to avoid them, authentication applications, electronic mail security, electronic transaction security and digital signatures, IP security, web security, system security, intruders and viruses, firewalls, introduction to cryptographic algorithms, standard security protocols, cyber crime, policy and regulations.

Reference Material:

1. Cryptography and Network Security: Principles and Practice, 4/E, William Stallings, Prentice Hall, 2005.
2. Government Policy documents on security issues.

Course Name: System Integration and Architecture

Course Structure: Lectures: 3 / Labs: 0 **Credit Hours:** 3

Prerequisites: Fundamentals of Information Technology (Required), Introduction to Software Development (Recommended)

Objectives:

This course will prepare the students to understand the system level requirements of an organization and acquire the required information and communication resources, integrate and deploy these resources in the form of a system.

Course Outline:

System level requirements gathering and analysis, acquisition, sourcing, integration, project management, testing and quality assurance, organizational context and architecture., intersystem's communication, data mapping and exchange, integrative coding, scripting techniques, software security and an overview of programming languages.

Suggested Text Books:

1. Enterprise Integration: An Architecture for Enterprise Application and Systems Integration (Paperback), by [Fred A. Cummins](#) (Author), Paperback: 496 pages, Publisher: Wiley; 1st edition (February 1, 2002), ISBN-10: 0471400106

BS IT Supporting Courses

Course Name: Technology Management

Course Structure: Lectures: 3/Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives:

- (a) to introduce basic management functions, focusing on technology management issues,
- (b) case study to appraise students real problems

Course Outline:

Introduction and issues in technology management; Basic management functions (Planning, Control, Decision making, organizing etc.); Business Change and Technology challenges and issues; Technology strategy, goals and objectives, common hurdles; Technology transfer issues related to hardware, software, communications, human resources, etc.; IT as change enabling technology, assessment and selection of technology, training planning, equipment and systems acquisition processes; Implementation processes; Common challenges in change management; Small case study.

Reference Material:

1. Robins Stephan, "Management"
2. Griffwn, "Principles of Management"
3. Robert Williams and Marks Walla. "The Ultimate Window 2000 System Administration's Guide.

Course Name: Organizational Behaviour

Course Structure: Lectures: 3/Labs: 0 **Credit Hours:** 3

Prerequisites: None

Objectives:

- (a) To introduce organizational behaviour and its impact on work within organization;
- (b) Impact of IT on individual behaviour.

Course Outline:

Introduction to Behavioural Science, an organizational behaviour, individual behaviour, personality, perceptions and attitudes, learning and reinforcement, motivation, team behaviour and organization, team dynamics and paradigms, leadership, organizational structure, organizational design, job design, stress and work, work processes and control issues, DM and its implications, communication effectiveness, performance and rewards, negative forces and conflict management, change issues, impact of IT on behaviour, power and politics in organizations.

Reference Material:

1. Organizational Behaviour: an Introductory Text, Huczinsky and Buchanan
2. Organizational Behaviour by Fred Luthans

Course Name: Information Systems**Course Structure:** Lectures: 3/Labs: 0 **Credit Hours:** 3**Prerequisites:** None**Objectives:**

Major emphasis than is usual for Information Systems analysis, design, and success and management aspects will be placed in order to discuss the management of the technical processes involved. Actual Case Studies will be central to the delivery of the unit. Recent, well-accepted, developments in all aspects of Information Systems development will also be covered and discussed. This course will facilitate students to understand the advanced concepts of information systems.

Course Outline:

Introduction and Classification of Information Systems, Lifecycle of IS Projects, Major Taxonomies of Information Systems, IS Strategies, Types of IS Strategies, Business Strategies and Types, Alignment of both Strategies, Information Systems success and Failure, Critical Success Factors, Information Systems Project Evaluation, IS Feasibility Study and Types, Managing Information Systems Projects, Structure of IS Projects, Managing Conflicts in Information Systems projects, Role of CIO, System Analysis of IS Projects, Design Issues in IS, Coupling, Cohesion and Structured Charts, Team Composition, Detailed IS Design Issues, Advanced Design Issues, Measuring Project Complexity, Prototype Approaches, CASE Tools, Soft System Methods (SSM), Rapid Application Development (RAD), Case Studies.

Reference Material:

1. Enterprise Information Systems, O'Brien and Marakas, 13th Edition, McGraw-Hill, 2007

BS IT Elective Courses**Course Name: Communication Systems Design****Course Structure:** Lectures: 3/Labs: 0 **Credit Hours:** 3**Prerequisites:** None**Objectives:**

The objective of this course is to learn theory of communication system design.

Course Outline:

Preliminaries on Deterministic and Random Signals, Characterization of Transmission Media and Devices, Analog Modulation Systems, Digital Modulation Systems, Digital Transmission of Analog Signals, Transmission over Dispersive Channels, Elements of Information Theory, Source and Channel Coding

Suggested Textbook:

1. Communication Systems: Fundamentals and Design Methods, [Nevio Benvenuto](#), [Roberto Corvaja](#), [Tomaso Erseghe](#), [Nicola Laurenti](#)
2. Communication System Design Using DSP Algorithms: With Laboratory Experiments for the TMS320C6713 DSK (Information Technology: Transmission, Processing and Storage) (Spiral-bound), by [Steven A. Tretter](#)

Course Name: Information Security**Course Structure:** Lectures: 3/Labs: 0 **Credit Hours:** 3**Prerequisites:** Computer Communication and Network**Objective:**

This course provides a broad overview of the threats to the security of information systems, the responsibilities and basic tools for information security, and the levels of training and expertise needed in organizations to reach and maintain a state of acceptable security. It covers concepts and applications of system and data security. Areas of particular focus include secure network design, implementation and transition issues, and techniques for responding to security breaches.

Course Outline:

Information Security Attacks & Vulnerabilities, Anatomy of Attack, Awareness and Management Commitment to Security, Security Policy, Information Security Network Architecture Design Rules, Rules for Selecting Security Hardware & Software, Physical Security Rules, Network Hardware Security, Operating System Security Rules, PC Operating Security Rules, Internet Security Rules, Application Security Rules, Software Validation and Verification Rules, Data Encryption Rules, Configuration Management Rules, Network Monitoring Rules, Maintenance and Troubleshooting Security Rules, Emergency Rules Attacks, An introduction to confidentiality, integrity, availability; authentication technologies and models, Controls and protection models, Security kernels, Secure programming, Information Auditing, Intrusion detection and response, Operational security issues, Physical security issues, Personnel security, Policy formation and enforcement, Access controls, Information flow, Legal, privacy and social issues, Identification and authentication in local and distributed systems; classification and trust modelling, Risks and vulnerabilities, Risk assessment, Database security, Encryption, Host-based and network-based security issues, Areas of particular focus include secure network design, implementation and transition issues, and techniques for responding to security breaches.

Reference Material:

1. Information Security Best Practices by George L. Stefanek, 2006.

Course Name: Communication Technologies**Course Structure:** Lectures: 3/Labs: 0 **Credit Hours:** 3**Prerequisites:** None

Objective:

Goals for the course include developing teaching strategies consistent with the constructivist philosophy of education that help new learners understand: how science & communication technology relate to society and the environment, how to use the processes of scientific inquiry and communication technological design, basic concepts from the major fields of science & communication technology. Integral to the course is our objective to help student-teachers develop their commitment to students and student learning; furthering professional knowledge through ongoing professional learning; and the application of professional knowledge to professional practice and leadership in learning communities.

Course Outline:

Introduction to Science & Technology, Interrelating Science, Communication Technology, Society and the Environment (STSE), Learning through Science & Technology, Communicating Science & Technology, Assessment for Learning in Science, Communication Technology, Science & Communication Technology for all Learners, Cross Curricular Connections, New Directions for Science & Technology Education, Maintaining Safe Learning Environments for Science & Communication Technology, Advance Topics in Communication Technology.

Reference Text:

1. Rees, C. and Halpern J. (2008) Readings for Science & Communication Technology.
2. Simon Haykin 4th Edition, Communication Systems.

Course Name: Database Management

Course Structure: Lectures: 2/Labs: 3 **Credit Hours:** 3

Prerequisites: Introduction to Database Systems

Objectives:

- (a) to manage large database systems,
- (b) to monitor the processing of database system.

Course Outline:

Advanced Structurejd Query Language (SQL): Complex Integrity Constraints (Assertions), Views in SQL, Designing and managing Triggers, Stored Procedures.

Database Security and Authorization: Discretionary Access Control; Mandatory Access Control; Role-based Access Control; Encryption and Public Key Infrastructures.

Database Tuning: File Structures and organizations; Hashing and Indexing; Database Workloads; Physical Design and Tuning Decisions; Index Selection; Tuning Schema: De-normalization and Decompositions; Tuning Queries and Views.

Recovery Techniques: Database backup and recovery from catastrophic failures.

Database System Architectures: Centralized and Client-Server Architectures; Parallel and Distributed Database Systems; Fragmentation and Replication; Distributed Catalogue Management.

Reference Material:

Latest editions of

1. R. Elmasri and S. Navathe. Fundamentals of Database Systems, 3rd Edition 2000, Benjamin/Cummings.
2. Abraham Silberschatz, Henry F. Korth S. Sudarshan. "Database System Concepts", 3rd Edition.
3. Raghu Ramakrishnan and Johannes Gehrke. "Database Management Systems", McGraw Hill, Third Edition.
4. T.Connolly and C.Begg . "Database Systems, a Practical Approach to Design, Implementation and Management", Pearson education, Third Edition.

Course Name: Knowledge-Based Systems

Course Structure: Lectures: 3/Labs: 1 **Credit Hours:** 4

Prerequisites: Fundamentals of Artificial Intelligence or Students should be familiar with programming and be able to work with elementary logic in propositional and predicate calculus environments.

Objectives:

- (a) to understand important problems, challenges, concepts and techniques from the field of Knowledge-Based Systems.
- (b) to learn how to analyze, design, and build systems with ability to deal with knowledge in various forms.
- (c) to know importance of an explanation of many systems' suggestions in a format accessible to humans.

Course Outline: Introduction to knowledge-based systems, Logic and automatic reasoning (forward and backward reasoning), Knowledge representation and reasoning models, Bayesian inference and other models of reasoning and decision making under uncertainty, Software lifecycle in knowledge-based systems, Rule-based expert systems, Architecture of a knowledge-based system, Feasibility analysis, Requirements specification and design, Knowledge acquisition and system implementation, Verification and validation.

Suggested Text Books:

1. Introduction to Expert Systems (3rd Edition) by Peter Jackson; Addison-Wesley Longman Publishing Company; 3rd Edition (January 1999).
2. Expert Systems: Principles and Programming by Joseph C. Giarratano, Gary D. Riley; PWS Publishing Co.; 4th Edition, Thomson/PWS Publishing Company; or latest edition.

Reference Material:

1. Artificial Intelligence: A Modern Approach (Latest Edition) by Stuart Russel, Peter Norvig, Prentice Hall.
2. Artificial Intelligence: Structures and Strategies for Complex Problem Solving by George F. Luger, Benjamin/Cummings Publishing.
3. The Engineering of Knowledge-Based Systems by A. Gonzalez and D. Dankel; 2nd Edition (Preprint), Prentice Hall, 2004.

4. Principles of Expert Systems, by P. Lucas and L. van der Gaog, Addison-Wesley, 1991 or latest ed.
5. A Guide to Expert Systems, by D.A. Waterman, Addison-Wesley, 1986 or latest edition.
6. Artificial Intelligence, by P.H. Winston, 3rd Edition, Addison-Wesley, 1992 or latest edition.

Course Name: Data Warehousing

Course Structure: Lectures: 2/Labs: 3 **Credit Hours:** 3

Prerequisites: Introduction to Database Systems

Objectives:

- (a) to manage large database systems,
- (b) to monitor the processing of database system.

Course Outline:

Introduction of the business context for data warehousing and decision support systems. Differences between TPS and DSS environments. Data warehouse Architecture. Data Marts. Differentiate Data Marts and Data Warehouse. Evaluation of Data Warehouse. Data Warehouse Design Methodology: Entity Relationship Modeling and Dimensional Modeling. OLAP in data warehousing and different types of OLAP such as MOLAP ROLAP and HOLAP. Indexing techniques used in data warehousing. Hardware and software systems consideration for data warehousing. Data warehouse maintenance.

Reference Material:

1. Paulraj Ponniah, Data Warehousing Fundamentals, John Wiley & Sons Inc., NY.
2. W.H. Inmon, Building the Data Warehouse (Second Edition), John Wiley & Sons Inc., NY.
3. Ralph Kimball and Margy Ross, The Data Warehouse Toolkit (Second Edition), John Wiley & Sons Inc., NY.

Course Name: Information Retrieval

Course Structure: Lectures: 3/Labs: 0 **Credit Hours:** 3

Prerequisites: Data Structures and Algorithms

Objectives:

- (a) to introduce basic management functions, focusing on technology management issues,
- (b) case study to appraise students real problems

Course Outline:

Basic and advanced techniques for text-based information systems: efficient text indexing; Basic IR Models: Boolean and vector-space retrieval models; ranked retrieval; text-similarity metrics; TF-IDF (term frequency, inverse document frequency); cosine similarity; Experimental Evaluation of IR: Performance metrics: recall, precision, and F-measure; Evaluations on benchmark text collections; Web search including crawling, link-based algorithms, and Web metadata; text/Web clustering, classification; text mining .

Reference Material:

1. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze (2008): Introduction to Information Retrieval, Cambridge University Press.
2. Berthier Ribeiro-Neto, and Ricardo Baeza-Yates (1999): Modern Information Retrieval, Addison-Wesley

COURSE CONTENTS FOR MS IT

Course Name: Advanced Database Management Systems

Course Structure: Lectures: 3/Labs: 0 **Credit Hours: 3**

Prerequisites: Introduction to Database Systems

Course Outline:

Object-Oriented Databases, Object-Relational Databases, Mobile Databases, Temporal, Spatial and Geographic Databases, Distributed Database Design, Distributed Multimedia Database Systems, Data Warehouse and OLAP Systems, Business Intelligence, XML Data Models, XML Documents and DTD, XML Query Languages, Current Research and Development Trends of Database Analysis, Design, Modeling and Applications.

Recommended Text:

1. An Advanced Course in Database Systems: Beyond Relational Databases, S. W. Dietrich and S. D. Urban, Prentice Hall, 2005.

Course Name: Telecom Management

Course Structure: Lectures: 3/Labs: 0 **Credit Hours: 3**

Prerequisites: None

Objectives:

The course provides the understanding of the operation and management of a telecommunication business.

Course Contents:

Introduction Information Technology: Computer Hardwar, Computer Software, Telecommunications and Networking, The Data Resource; Applying Information Technology: Enterprise Systems, Managerial Support Systems, E-Business Systems; Acquiring Information Systems: Basic Systems Concepts and Tools, Methodologies for Custom Software Development, Methodologies for Purchased Software Packages, IT Project Management, Supporting Computer Users, The

Information Management System: Planning Information Systems Resources, Leading the Information Systems Function, Information Security, Legal, Ethical, and Social Issues, Extensive Case Studies in each topic discussed in the course.

Reference Material:

1. Managing Information Technology (6th Edition), by [Carol V Brown](#) (Author), [Daniel W. DeHayes](#) (Author), [Jeffrey A. Hoffer](#) (Author), [Wainright E. Martin](#) (Author), [William C Perkins](#)
2. Managing Information Technology: What Managers Need to Know by [Carol V Brown](#), [Jeffrey A. Hoffer](#), [Daniel W. DeHayes](#), [Wainright E. Martin](#), [William C Perkins](#)

Course Name: Information Security and Assurance

Course Structure: Lectures: 3/Labs: 0 **Credit Hours:** 3

Prerequisites: Network Security

Objective:

This course explores the issues of ethical challenges and legal issues that fact security practitioners. Understanding and evaluation the impact of legal and ethical issues on information security practice, privacy and security laws and regulations and assurance such as HIPAA, GLBA, Sarbanes-Oxley, Patriot Act, FISMA, CISRA and other. Techniques for planning, managing and implementing strategies based on these regulatory requirements will be discussed.

The protection of information assets underpins the commercial viability and profitability of all enterprises and the effectiveness of public sector organizations. Information security should not be left to chance but should be managed to ensure it provides efficient and effective safeguards for your organization's information assets.

Course Outline:

Information Assurance, Historical Approaches to Information Security and Information Assurance, Define the System Boundaries, Perform Vulnerability and Threat Analyses, Implement Threat Control Measures, Very Effectiveness of Threat Control Measures, Conduct Accident/Incident Investigations.

Information Assurance Requirement in Modern Information Systems, Identification of Basic Services of Security e.g. Confidentiality, Integrity authentication, non-repudiation and digital signatures, Intrusion detection, Introduction to Conventional and Un-Conventional Cryptosystems, Security Mechanisms such as Hashing, Biometrics, Smartcards etc, Security Protocols for End-to-End Secure Communication on all Types of Networks, Security Policies, Standards and Auditing, Legal, Ethical, and Professional Issues in Information Security

Recommended Text:

1. M. Whitman & H. Mattord (2003). *Principles of Information Security*. Course Technology, ISBN: 0619063181
2. M. D. Abrams, S. Jajodia, and H. J. Podell, eds. (1995). *Information Security: An Integrated Collection of Essays*, IEEE Computer Society Press, online at <http://www.acsac.org/secshelf/book001/book001.html>

3. Bruce Schneier (2002). *Secrets & Lies: Digital Security in a Networked World*, Counterpane Internet Security, ISBN: 0-471-25311-1

Reference Text:

1. Information Security Management Handbook By Harold F. Tipton, Micki Krause
2. Information Assurance and Computer Security By Johnson P.Thomas, Mohamed Essaaidi
3. Computer Security Assurance using the common criteria by Merkow & Breithaupt
4. Practical Guide to Security Engineering and Information Assurance by Debra S.Herrmann

Course Name: Information Technology Infrastructure

Course Structure: Lectures: 3/Labs: 0 **Credit Hours:** 3

Prerequisites: None

Course Outline:

IT strategy and management, strategic planning for IT, IT investment and valuation, business and information technology strategy linkage, implementation of service strategies, and risks and critical success factors, Development and maintenance of information technology policies, documents, and architectures for the design of IT service solutions/processes, Service design objectives, Selecting the model, risk analysis; implementation; cost; and control & measurement, long term change and release management concepts and practices, cultural and organizational change management; knowledge management (KM); control & measurement; and tools & methods, change management, application management; scalability; control & measurement; enterprise information infrastructure, IT infrastructure for virtual organizations, State of IT governance

Reference Material:

1. Global Information Infrastructure: The Birth, Vision, and Architecture, Andrew S. Targowski, 1996, IGI
2. Information Technology Governance and Service Management, Aileen Cater-Steel 2009.
3. Best Practice for ICT Infrastructure Management, Office of Government Commerce, Edition: 6, illustrated, Published by The Stationery Office, 2002

MS IT Elective Courses

Course Name: Distributed Databases

Course Structure: Lectures: 3/Labs: 0 **Credit Hours: 3**

Prerequisites: Introduction to Database Systems

Objectives:

Students will learn the usage of different design strategies for distributed databases, and will study query processing techniques as well as transaction management and concurrency control concepts used in such systems

Course Outline:

Introduction to Distributed Data Processing; Distributed DBMS Architecture; Distributed Database Design: Issues, Fragmentation and Allocation; Integrity Constraints, Distributed Query Processing; Query Decomposition and Data Localization; Query Optimization; Distributed Transaction Management and Concurrency Control; Distributed DBMS Reliability and Replication Techniques; Multidatabase Systems.

Reference Material:

1. M.T. Ozsü, P. Valduriez (eds.): Principles of Distributed Database Systems (2nd Edition), Prentice Hall, 1999
2. P. Bernstein and E. Newcomer, Principles of Transaction Processing. Morgan Kaufmann, 1997
3. M. Buretta, Data Replication. Wiley, 1997
4. R. Elmasri and S. Navathe. Fundamentals of Database Systems, Benjamin/Cummings.

Course Name: Data Mining

Course Structure: Lectures: 3/Labs: 0 **Credit Hours: 3**

Prerequisites:

Objectives:

- (a) to introduce the techniques, tools and applications of data mining,
- (b) to apply DM techniques to a variety of research and application projects.

Course Outline:

Introduction to Data Mining (DM), High-Dimensional Data; Classifiers; Decision Trees; Neural Networks; Clustering Algorithms; Association Rules & Sequences; Commercially-Available DM Tools: Excel. Teradata. SAS. SPSS. IBM. Oracle. Whitecross. The CRISP-DM process.

Reference Material:

1. David Hand, Heikki Mannila and Padhraic Smyth. "Principles of Data Mining".
2. Pub. Prentice Hall of India.
3. Sushmita Mitra and Tinku Acharya. "Data Mining: Multimedia, Soft Computing and Bioinformatics". Pub. Wiley and Sons Inc.
4. Usama M. Fayyad et al. "Advances in Knowledge Discovery and Data Mining",
5. The MIT Press.
6. Richard Roiger & Michael Geatz. "Data Mining: A Tutorial –Based Primer", Addison-Wesley.

Course Name: Advanced Topics in Databases**Course Structure:** Lectures: 3/Labs: 0 **Credit Hours: 3****Prerequisites:** Advanced Database Management**Objectives:**

In recent years, there has been an explosion of information in a variety of environments that pose significantly different data management challenges than traditional database domains. Examples include semantic heterogeneity, sensor networks, World Wide Web, scientific domains, XML, P2P networks etc. This course is a combination of various advanced topics. The aim of this course is to explore the latest techniques, trends, ideas, and what are involved in designing and evaluating the cutting-edge database technologies.

Course Outline:

This course is intended to be highly interactive. The main activity of the lectures will be discussions based around a set of papers. All students are required to read technical papers, to answer specific questions, and to prepare new questions prior to class discussions. In addition, each student is required to lead the discussion on one or two of these technical papers.

Recommended topics include, but are not limited to, the following:

- Data integration.
- Semantic heterogeneity.
- Ontology engineering.
- Data caching and replication.
- Streaming data.
- Semi-structured data (i.e., XML) storage.
- Mapping between XML and relational databases.
- Pervasive and mobile distributed database management

Reference Material:

1. Research Papers form HEC Digital Library.

Course Name: Information Technology Architecture**Course Structure:** Lectures: 3 / Labs: 0 **Credit Hours: 3****Prerequisites:** System Integration and Architecture

Objectives:

Objective of this course is to understand the Information Technology Architecture as a framework and a set of strategies for the utilization and management of information technology, composed of principles, policies, and standards that guide the engineering of an organization's IT systems and infrastructure in a way that ensures alignment with business needs. Students will be able to select and implement the computing platforms, software, networks, and related products that interconnect different systems and ensure their interoperability.

Course Outline:

Business Architecture: Business Strategy, Business Support Functions and Processes; Information Architecture: Information Needs, Information Management Processes; Application Architecture: Guidelines for Design and Development of Business Applications, Policies, Standards, and Tools for Application Development; Infrastructure Architecture: Hardware, Software, and Communication Network for Information Storage, Transfer, Processing, Management; Security Architecture: Security Services, Security Framework; IT Management and Governance: Planning, Decision Making, Follow up, Assessment

Reference Material:

1. Enterprise Integration: An Architecture for Enterprise Application and Systems Integration (Paperback), by [Fred A. Cummins](#) (Author), Paperback: 496 pages, Publisher: Wiley; 1st edition (February 1, 2002), ISBN-10: 0471400106
2. Building Enterprise Information Architectures: Reengineering Information Systems, Melissa A. Cook
3. Constructing Blueprints for Enterprise IT Architectures, Benard H. Boar
4. Enterprise Architecture Planning, Steven H. Spewak, Steven C. Hill
5. Enterprise-wide IT Architecture: <http://www.ewita.com/>
6. The Open Group: <http://www.opengroup.org/itac/>

COMPULSORY COURSES

COMPULSORY COURSES IN ENGLISH FOR BS (4 YEAR) IN BASIC & SOCIAL SCIENCES

English I (Functional English)

Objectives:

Enhance language skills and develop critical thinking.

Course Contents

Basics of Grammar
Parts of speech and use of articles
Sentence structure, active and passive voice
Practice in unified sentence
Analysis of phrase, clause and sentence structure
Transitive and intransitive verbs
Punctuation and spelling

Comprehension

Answers to questions on a given text

Discussion

General topics and every-day conversation (topics for discussion to be at the discretion of the teacher keeping in view the level of students)

Listening

To be improved by showing documentaries/films carefully selected by subject teachers

Translation skills

Urdu to English

Paragraph writing

Topics to be chosen at the discretion of the teacher

Presentation skills

Introduction

Note: Extensive reading is required for vocabulary building

Recommended books:

1. **Functional English**
 - a) Grammar
 1. Practical English Grammar by A.J. Thomson and A.V. Martinet. Exercises 1. Third edition. Oxford University Press. 1997. ISBN 0194313492
 2. Practical English Grammar by A.J. Thomson and A.V. Martinet. Exercises 2. Third edition. Oxford University Press. 1997. ISBN 0194313506

- b) Writing
 1. Writing. Intermediate by Marie-Christine Boutin, Suzanne Brinand and Françoise Grellet. Oxford Supplementary Skills. Fourth Impression 1993. ISBN 0 19 435405 7 Pages 20-27 and 35-41.
- c) Reading/Comprehension
 1. Reading. Upper Intermediate. Brian Tomlinson and Rod Ellis. Oxford Supplementary Skills. Third Impression 1992. ISBN 0 19 453402 2.
- d) Speaking

English II (Communication Skills)

Annexure-B

Objectives:

Enable the students to meet their real life communication needs.

Course Contents

Paragraph writing

Practice in writing a good, unified and coherent paragraph

Essay writing

Introduction

CV and job application

Translation skills

Urdu to English

Study skills

Skimming and scanning, intensive and extensive, and speed reading, summary and précis writing and comprehension

Academic skills

Letter/memo writing, minutes of meetings, use of library and internet

Presentation skills

Personality development (emphasis on content, style and pronunciation)

Note: documentaries to be shown for discussion and review

Recommended books:

Communication Skills

- a) Grammar
 1. Practical English Grammar by A.J. Thomson and A.V. Martinet. Exercises 2. Third edition. Oxford University Press 1986. ISBN 0 19 431350 6.
- b) Writing
 1. Writing. Intermediate by Marie-Christine Boutin, Suzanne Brinand and Françoise Grellet. Oxford Supplementary Skills. Fourth Impression 1993. ISBN 019 435405 7 Pages 45-53 (note taking).
 2. Writing. Upper-Intermediate by Rob Nolasco. Oxford Supplementary Skills. Fourth Impression 1992. ISBN 0 19 435406 5 (particularly good

for writing memos, introduction to presentations, descriptive and argumentative writing).

- c) Reading
1. Reading. Advanced. Brian Tomlinson and Rod Ellis. Oxford Supplementary Skills. Third Impression 1991. ISBN 0 19 453403 0.
 2. Reading and Study Skills by John Langan
 3. Study Skills by Riachard Yorkey.

English III (Technical Writing and Presentation Skills)

Annexure-C

Objectives:

Enhance language skills and develop critical thinking

Course Contents

Presentation skills

Essay writing

Descriptive, narrative, discursive, argumentative

Academic writing

How to write a proposal for research paper/term paper

How to write a research paper/term paper (emphasis on style, content, language, form, clarity, consistency)

Technical Report writing

Progress report writing

Note: Extensive reading is required for vocabulary building

Recommended books:

Technical Writing and Presentation Skills

- a) Essay Writing and Academic Writing
1. Writing. Advanced by Ron White. Oxford Supplementary Skills. Third Impression 1992. ISBN 0 19 435407 3 (particularly suitable for discursive, descriptive, argumentative and report writing).
 2. College Writing Skills by John Langan. Mc=Graw-Hill Higher Education. 2004.
 3. Patterns of College Writing (4th edition) by Laurie G. Kirsznar and Stephen R. Mandell. St. Martin's Press.
- b) Presentation Skills
- c) Reading
- The Mercury Reader. A Custom Publication. Compiled by norther Illinois University. General Editors: Janice Neulib; Kathleen Shine Cain; Stephen Ruffus and Maurice Scharton. (A reader which will give students exposure to the best of twentieth century literature, without taxing the taste of engineering students).

Pakistan Studies (Compulsory)

Introduction/Objectives

- Develop vision of historical perspective, government, politics, contemporary Pakistan, ideological background of Pakistan.
- Study the process of governance, national development, issues arising in the modern age and posing challenges to Pakistan.

Course Outline

1. Historical Perspective

- a. Ideological rationale with special reference to Sir Syed Ahmed Khan, Allama Muhammad Iqbal and Quaid-i-Azam Muhammad Ali Jinnah.
- b. Factors leading to Muslim separatism
- c. People and Land
 - i. Indus Civilization
 - ii. Muslim advent
 - iii. Location and geo-physical features.

2. Government and Politics in Pakistan

Political and constitutional phases:

- a. 1947-58
- b. 1958-71
- c. 1971-77
- d. 1977-88
- e. 1988-99
- f. 1999 onward

3. Contemporary Pakistan

- a. Economic institutions and issues
- b. Society and social structure
- c. Ethnicity
- d. Foreign policy of Pakistan and challenges
- e. Futuristic outlook of Pakistan

Books Recommended

1. Burki, Shahid Javed. *State & Society in Pakistan*, The Macmillan Press Ltd 1980.
2. Akbar, S. Zaidi. *Issue in Pakistan's Economy*. Karachi: Oxford University Press, 2000.
3. S.M. Burke and Lawrence Ziring. *Pakistan's Foreign policy: An Historical analysis*. Karachi: Oxford University Press, 1993.
4. Mehmood, Safdar. *Pakistan Political Roots & Development*. Lahore, 1994.
5. Wilcox, Wayne. *The Emergence of Banglades.*, Washington: American Enterprise, Institute of Public Policy Research, 1972.
6. Mehmood, Safdar. *Pakistan Kayyun Toota*, Lahore: Idara-e-Saqafat-e-Islamia, Club Road, nd.
7. Amin, Tahir. *Ethno - National Movement in Pakistan*, Islamabad: Institute of Policy Studies, Islamabad.

8. Ziring, Lawrence. *Enigma of Political Development*. Kent England: WmDawson & sons Ltd, 1980.
9. Zahid, Ansar. *History & Culture of Sindh*. Karachi: Royal Book Company, 1980.
10. Afzal, M. Rafique. *Political Parties in Pakistan*, Vol. I, II & III. Islamabad: National Institute of Historical and cultural Research, 1998.
11. Sayeed, Khalid Bin. *The Political System of Pakistan*. Boston: Houghton Mifflin, 1967.
12. Aziz, K.K. *Party, Politics in Pakistan*, Islamabad: National Commission on Historical and Cultural Research, 1976.
13. Muhammad Waseem, *Pakistan Under Martial Law*, Lahore: Vanguard, 1987.
14. Haq, Noor ul. *Making of Pakistan: The Military Perspective*. Islamabad: National Commission on Historical and Cultural Research, 1993.

ISLAMIC STUDIES (Compulsory)

2 (2-0)

Objectives:

This course is aimed at:

- 1 To provide Basic information about Islamic Studies
- 2 To enhance understanding of the students regarding Islamic Civilization
- 3 To improve Students skill to perform prayers and other worships
- 4 To enhance the skill of the students for understanding of issues related to faith and religious life.

Course Outlines

Introduction to Quranic Studies

- 1) Basic Concepts of Quran
- 2) History of Quran
- 3) Uloom-ul -Quran

Study of Selected Text of Holly Quran

- 1) Verses of Surah Al-Baqra Related to Faith (Verse No-284-286)
- 2) Verses of Surah Al-Hujrat Related to Adab Al-Nabi (Verse No-1-18)
- 3) Verses of Surah Al-Mumanoon Related to Characteristics of faithful (Verse No-1-11)
- 4) Verses of Surah al-Furqan Related to Social Ethics (Verse No.63-77)
- 5) Verses of Surah Al-Inam Related to Ihkam (Verse No-152-154)

Study of Selected Text of Holly Quran

- 1) Verses of Surah Al-Ihzab Related to Adab al-Nabi (Verse No.6,21,40,56,57,58.)
- 2) Verses of Surah Al-Hashar (18,19,20) Related to thinking, Day of Judgment
- 3) Verses of Surah Al-Saf Related to Tafakar, Tadabar (Verse No-1,14)

Seerat of Holy Prophet (S.A.W) I

- 1) Life of Muhammad Bin Abdullah (Before Prophet Hood)
- 2) Life of Holy Prophet (S.A.W) in Makkah
- 3) Important Lessons Derived from the life of Holy Prophet in Makkah

Seerat of Holy Prophet (S.A.W) II

- 1) Life of Holy Prophet (S.A.W) in Madina
- 2) Important Events of Life Holy Prophet in Madina
- 3) Important Lessons Derived from the life of Holy Prophet in Madina

Introduction To Sunnah

- 1) Basic Concepts of Hadith
- 2) History of Hadith
- 3) Kinds of Hadith
- 4) Uloom –ul-Hadith
- 5) Sunnah & Hadith
- 6) Legal Position of Sunnah

Selected Study from Text of Hadith

Introduction To Islamic Law & Jurisprudence

- 1) Basic Concepts of Islamic Law & Jurisprudence
- 2) History & Importance of Islamic Law & Jurisprudence
- 3) Sources of Islamic Law & Jurisprudence
- 4) Nature of Differences in Islamic Law
- 5) Islam and Sectarianism

Islamic Culture & Civilization

- 1) Basic Concepts of Islamic Culture & Civilization
- 2) Historical Development of Islamic Culture & Civilization
- 3) Characteristics of Islamic Culture & Civilization
- 4) Islamic Culture & Civilization and Contemporary Issues

Islam & Science

- 1) Basic Concepts of Islam & Science
- 2) Contributions of Muslims in the Development of Science
- 3) Quranic & Science

Islamic Economic System

- 1) Basic Concepts of Islamic Economic System
- 2) Means of Distribution of wealth in Islamic Economics
- 3) Islamic Concept of Riba
- 4) Islamic Ways of Trade & Commerce

Political System of Islam

- 1) Basic Concepts of Islamic Political System
- 2) Islamic Concept of Sovereignty
- 3) Basic Institutions of Govt. in Islam

Islamic History

- 1) Period of Khlaft-E-Rashida
- 2) Period of Ummayyads
- 3) Period of Abbasids

Social System of Islam

- 1) Basic Concepts of Social System of Islam
- 2) Elements of Family
- 3) Ethical Values of Islam

Reference Books:

- 1) Hameed ullah Muhammad, "Emergence of Islam", IRI, Islamabad
- 2) Hameed ullah Muhammad, "Muslim Conduct of State"
- 3) Hameed ullah Muhammad, "Introduction to Islam"
- 4) Mulana Muhammad Yousaf Islahi,"
- 5) Hussain Hamid Hassan, "An Introduction to the Study of Islamic Law" leaf Publication Islamabad, Pakistan.
- 6) Ahmad Hasan, "Principles of Islamic Jurisprudence" Islamic Research Institute, International Islamic University, Islamabad (1993)

- 7) Mir Waliullah, “Muslim Jrisprudence and the Quranic Law of Crimes” Islamic Book Service (1982)
- 8) H.S. Bhatia, “Studies in Islamic Law, Religion and Society” Deep & Deep Publications New Delhi (1989)
- 9) Dr. Muhammad Zia-ul-Haq, “Introduction to Al Sharia Al Islamia” Allama Iqbal Open University, Islamabad (2001)