# CURRICULUM

# OF

# Computer Science, Software Engineering & Information Technology

(Revised 2004)


**HIGHER EDUCATION COMMISSION**
**ISLAMABAD**

# CURRICULUM DIVISION, HEC

| | |
|---|---|
| Prof. Dr. Altaf Ali G. Shaikh | Adviser (HRD) |
| Farman Ullah Anjum | Director Curriculum |
| Malik Ghulam Abbas | Deputy Director |
| Miss Ghayyur Fatima | Deputy Director |
| Mr. M. Tahir Ali Shah | Assistant Director |
| Mrs. Noshaba Awais | Assistant Director |

Composed by Mr. Zulfiqar Ali, HEC Islamabad

## *Contents*

## 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.

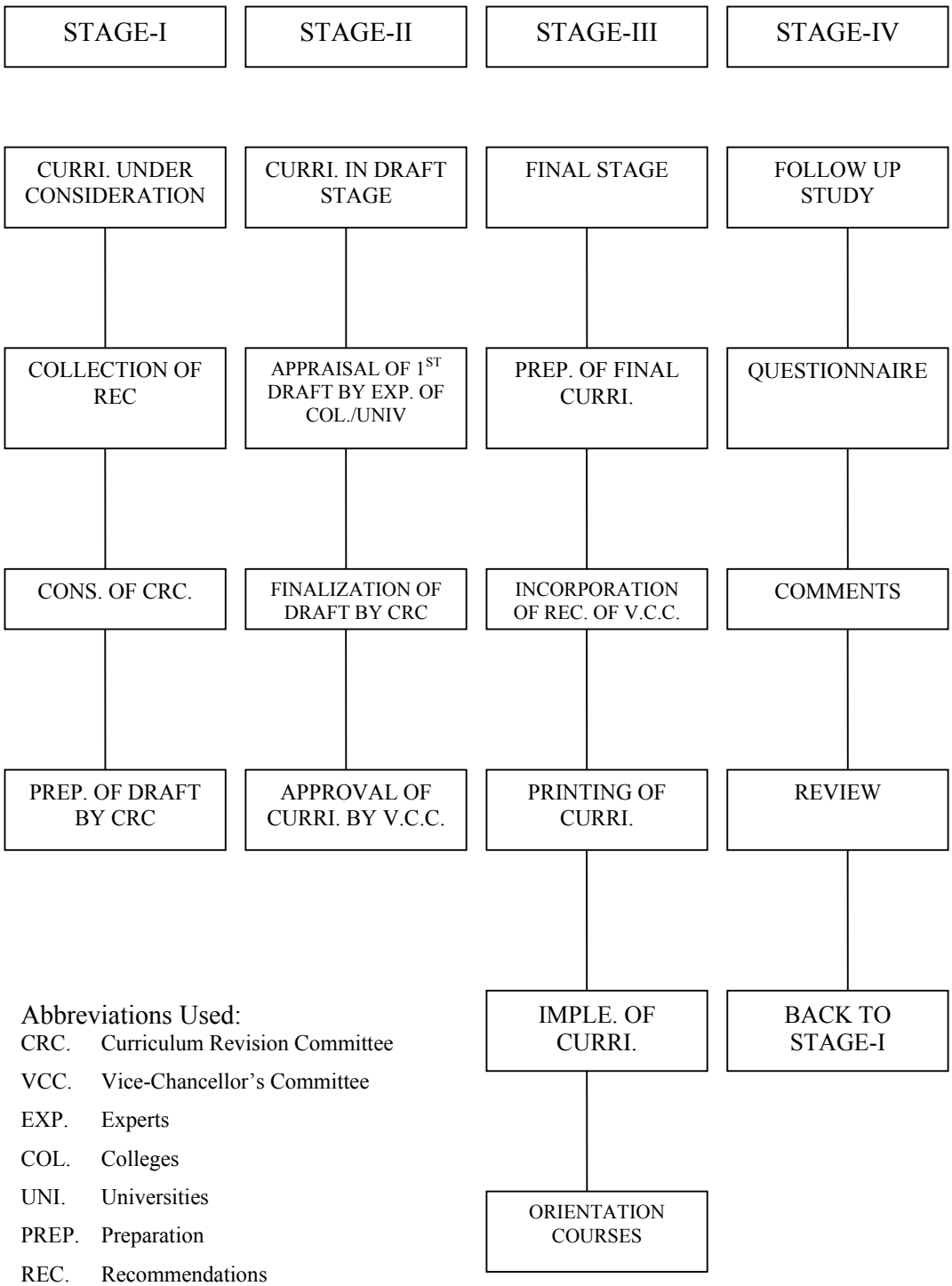
In exercise of the powers conferred by sub-section (1) of section 3 of the Federal Supervision of Curricula Textbooks and Maintenance of Standards of Education Act 1976, the Federal Government vide notification no. D773/76-JEA (Cur.), dated December 4, 1976, appointed University Grants Commission as the competent authority to look after the curriculum revision work beyond class XII at 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-Chancellors' Committee, curriculum of a subject must be reviewed after every 3 years. For the purpose, various committees are constituted at the national level comprising senior teachers nominated by universities. Teachers from local degree colleges and experts from user organizations, where required, are also included in these committees. The National Curriculum Revision Committees of Computer Science, Information Technology and Software Engineering in its meetings held in March, April & June 2004 respectively, at the HEC Secretariat, Islamabad and Regional Centre, Karachi revised the curricula after due consideration of the comments and suggestions received from universities and colleges where the subject under consideration is taught. The Joint meeting of the NCRC finalized the combined curricula for Computer Science, Software Engineering, and Information Technology. The final draft prepared by the National Curriculum Revision Committees duly approved by the Competent Authority is being circulated for implementation by the relevant institutions.

***(PROF. DR. ALTAF ALI G. SHAIKH)***
***Adviser (HRD)***

August 2004

# CURRICULUM DEVELOPMENT




Abbreviations Used:

CRC. Curriculum Revision Committee

VCC. Vice-Chancellor's Committee

EXP. Experts

COL. Colleges

UNI. Universities

PREP. Preparation

REC. Recommendations

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

## *I. Introduction*

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 (2003)
- National Curriculum Revision Committee — Software Engineering (2004)
- National Curriculum Revision Committee — Information Technology (2004)

All committees held their preliminary meetings to establish the respective first draft of curriculum. The reports delivered by theses 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 Ph.D. programmes.

All three committees 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.

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. To this end, it became essential to integrate the work of all three committees under the umbrella of Computing and to identify commonalities and differences among all three disciplines. 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 programmes:

1. Prof. Dr. Aftab Ahmad **Convener**
   Convener
   National Curriculum Revision Committee
   (Computer Science)
   Dean, Faculty of Engineering and Information Technology
   Foundation University
   Institute of Management and Computer Sciences
   Rawalpindi

2. Prof. Dr. Jamil Ahmad **Secretary**
   Secretary, National Curriculum Revision Committee
   (Computer Science)
   Dean, Iqra University
   Islamabad

3. Prof. Dr. Farhana Shah **Member**
   Convener
   National Curriculum Revision Committee
   (Information Technology)
   Director Institute of Information Technology &
   Chairperson Department of Computer Science
   Quaid-i-Azam University
   Islamabad

4. Prof. Dr. Aftab Mahrouf **Member**
   Convener
   National Curriculum Revision Committee
   (Software Engineering)
   Director, FAST-National University of
   Computer and Emerging Sciences
   Islamabad

5. Prof. Dr. Naveed Ikram **Member**
   Secretary, National Curriculum Revision Committee
   (Information Technology)
   Dean Faculty of Information Technology
   Riphah International University
   Islamabad

6. Prof. Dr. Imdad Ali Ismaili **Member**
   Secretary, National Curriculum Revision Committee
   (Software Engineering)
   Institute of Information Technology, University of Sindh
   Jamshoro

7. Prof. Dr. Nazir Ahmed Sangi **Member**
   Dean, Faculty of Sciences &
   Chairman Department of Computer Science
   Allama Iqbal Open University
   Islamabad

8. Prof. Dr. Khalid Rashid **Member**
   Dean, Faculty of Applied Sciences
   International Islamic University
   Islamabad

9. Prof. Dr. Jaffar-ur-Rahman **Member**
   Dean, Faculty of Engineering and Computer Science
   Muhammad Ali Jinnah University
   Islamabad

10. Prof. Dr. Muhammad Afzal **Member**
    Director, Centre of Information Technology
    University of Arid Agriculture
    Rawalpindi

11. Dr. Shoeb Ahmad Khan **Member**
    Associate Professor
    College of EME, NUST
    Rawalpindi

A two-day meeting of the above Committee was held on August 6-7, 2004 at Islamabad and all participants contributed significantly to establish a model to structure all degree programmes on the basis of Computing in a systematic manner.

Computing (A nucleus of all activities including technical, academic, professional and development practices relating computers) provides a wide range of choices on how an individual might focus his or her professional life. This report provides an overview of the different kinds of degree programmes in Computing that are currently available and for which curriculum standards are now available. It is believed that this report may be an essential source for university faculty, administrators, students, parents and professionals who need to be aware of Computing as a broad based discipline that crosses the boundaries between science, engineering, and professional practice. In reality, Computing consists of several disciplines. Various questions are naturally critical including: what are the different kinds of Computing degree programmes or how are they similar and how are they different? The variety of degree programmes in Computing presents prospective students, educators, and administrators with important choices where they may focus their efforts.

The following part of this section introduces the objectives of the report, the basic concept of Computing and a brief overview of Computer Science, Software Engineering and Information Technology disciplines and the proposed models and various proposed curricula structures are discussed here. The following sections of the report provide a complete detail of the proposed curricula pertaining to Computer Science, Software Engineering and Information Technology respectively regarding BS, and MS programs. For each discipline, all degree programs are presented with objectives, design and structures, courses objectives, course contents and recommended reference material. Generalized structure proposed for PhD for all three disciplines is presented at the end of this report.

Computing is a dynamic field and accordingly a good care has been taken to design a flexible structure that will maintain currency with the latest scientific and technological advancements in the field. Moreover, it seems that the Computing is a discipline that incorporates scientific, engineering, and creative features. A reasonable emphasis has been given to formal scientific and engineering areas to enhance the level of formalization in the proposed degree programmes. Technology can play an important role in the implementation of Computing programmes. As a result, all programmes are structured on essential dimensions including scientific knowledge, technology and design skills.

## *II. Objectives*

1. Computing will be one of the key factors driving in progress the 21st century — it will transform the way we live, learn, work, and play. Advances in Computing and its technologies will create a new infrastructure for business, commerce, manufacturing, communication, scientific research, education, and social interaction. This expanding infrastructure will provide us with new tools for communicating throughout the world and for acquiring knowledge and insight from information. It will provide a vehicle for economic growth.

2. Vigorous Computing education and research and development are essential for achieving our national aspirations of the 21st century. As we advance in the 21st century, the opportunities for innovation in Computing are larger than they have ever been — and more important. The technical advances that led to today's information tools, such as electronic computers and the Internet, are accessible with continuously decreasing cost.

3. The nation is in need of significant efforts on education and research in Computing and communication systems. ***If the results are to be available when needed, we must act now to reinvigorate the long-term Computing education and research endeavour and to revitalize the computing infrastructure at university campus level.***

We need to ensure that advances in Computing work should benefit us and that the majority of Pakistanis have the education and training needed to prosper in a world that will increasingly depend on Computing. The benefits of these transformations caused by Computing for our national future are extraordinary. A networked society can reach out to all its citizens, can bring us closer together and address many societal issues.

4. The proposed plan of Computing directly supports the education and preparation of our young people for careers in Computing research, and the training of workers who need to upgrade their skills to keep pace with a changing marketplace. Trained people are a major product of publicly supported research. These trained professionals are critical national human resource, and will create and develop new ideas, form a talent pool for existing business, and launch new companies. The realization of the positive transformations as newly designed degree programmes will be described in the next part of this report.

5. Now as the current world is considered a global village due to the rapid flow of information from one place to another, the one who can share and access this information is considered a part of the global village. The astronomical growth in Computing compels the whole professional world to reorient their efforts to maximize utilization of Computing in their professional activities. This enables all the educational institutes, which are primarily responsible to create trained manpower, for devising programmes that will lead to an optimum utilization of Computing in different spheres of life. There is a tremendous challenge to create well-equipped Computing professionals who have the ability and expertise to respond adequately to growing needs of the industry.

6. Realizing the high market demands and shortage of quality in Computing education at different levels an enhancement in existing Computing programmes is proposed. This report is based upon horizontal and vertical growth in those disciplines of Computing where a high demand is present and it is well estimated that this will grow universally for the years to come. Thus, it is high time for our universities to focus its resources together to seize a maximum share from this exponentially growing market.

7. The report conceptually and philosophically provides two-dimensional model of the overall Computing Educational Infrastructure. The concept nicely reflects national and international frontiers on Computing education for the upcoming future.

8. Our universities have quality human and technological resources and an excellent infrastructure. The report provides new horizons, strategies and challenges to transform the existing infrastructure into a leading Computing university.

9. Computing is a very dynamic field. It is essential that the curricula structures are dynamic accordingly and flexible to handle the latest scientific and technological advancements in the field.

## *III. Computing*

Computing is a dynamic, flexible and an integrated large domain of scientific and engineering knowledge, technologies, and research and development with enormous applications. However, it may be characterized as a nucleus of all activities including technical, academic, professional and development practices relating computers. Accordingly, it involves development of technologies and techniques via hardware, software, and communications. Moreover, innovative and limitless applications of Computing pertain to designing and building of hardware and software systems for a variety of purposes. Additionally, it deals with the automatic processing, protection, management and structuring of a whole range of information in different formats.

Computing is not just a single discipline but is a family of disciplines. There may be dozens if not hundreds around the world. However, among them, five appear to have some distinction today. These include the following:

- ***Computer Science***
- ***Computer Engineering***
- ***Information Systems***
- ***Information Technology***
- ***Software Engineering***

It may be pointed out here that all subsequent sections of the report will be restricted to the three disciplines of ***Computer Science, Software Engineering*** and ***Information Technology*** and the following is a brief overview of these disciplines:

### *Computer Science*

*Computer Science* spans a wide range, from its theoretical and algorithmic foundations to cutting-edge developments in robotics, computer vision, intelligent systems, bioinformatics, and other exciting areas. The overall scope of Computer Science may be viewed into the following three categories:

- To develop effective ways to solve computing problems. For example, Computer Science develops the best possible ways to store information in databases, send data over networks, and display complex images. The theoretical background offered by Computer Science allows determining the best performance possible, and their study of algorithms. It enables

to develop new problem-solving approaches that provide better performance.

- It devises new ways to use computers intelligently and effectively. Progress in the areas of networking, database, and human-computer-interface came together as a result of the world-wide-web, which changed the entire world. Now, researchers are working to make robots that are practical aides and demonstrate intelligence, databases that create new knowledge and, in general, use computers to do new things.

- It deals with the design and implementation of software systems. Computer Science provides training and skills for the successful implementation of software systems that solve challenging programming jobs. Computer Science spans the range from theory to models, design and programming. Computer Science offers a comprehensive foundation that permits graduates to adapt to new technologies and new ideas.

***Software Engineering***

Software Engineering is the discipline of developing and maintaining software systems that behave reliably and efficiently, and are affordable to develop and maintain. However, more recently it has evolved in response to the increased importance of software in safety-critical applications and to the growing impact of large and expensive software systems in a wide range of situations. The following describes an overview of Software Engineering:

- To provide software development practices requires more than just the underlying principles of computer science; it offers the rigor that the engineering disciplines bring to the reliability and trustworthiness of the artefacts.

- Software Engineering is different in character from other engineering disciplines, due to both the intangible nature of software and to the discontinuous nature of software operation.

- It seeks to integrate the science of Computer Science with the engineering principles developed for tangible and physical phenomena.

***Information Technology***

Information Technology refers to meet the technology needs of business, government, healthcare, schools, and other kinds of organizations.

The summary of the overview of Information Technology discipline is given by:

- It deals with system configuration and administration, computer and network hardware installation and maintenance.

- Use and management of databases; development and modelling; creation and management of websites and web-based systems; e-governance and e-commerce; digital voice and video communications; and computer and information security.

- To offer techniques that respond to practical everyday needs of business and other organizations.

## IV. A Two-Dimensional Model for Computing *Curricula Structures*

The major objective of a successful model of overall educational infrastructure of Computing should revolve around the mission of ***Maintaining Knowledge Currency in the 21st Century.***

The rapid pace of change of Computing offers various opportunities for existing and upcoming universities. The Computing Educational Infrastructure spans all forms of pedagogical activities for inducing its disciplines at all levels. Each individual who is related to Computing via education or industry will benefit from the advantage of vertical growth in his or her domain. The Computing Educational Infrastructure aims at attracting potential manpower that seeks to obtain quality education. The infrastructure is segregated into components on the basis of the nature of the manpower related to Computing.

Fresh graduates, business executives, Computing professionals, professionals from inter-disciplines and academicians will aid to ascertain the different components of the Computing infrastructure. The Computing educational infrastructure can be expressed as a two-dimensional model as described below.

The model is flexible with the prospective changes and trends in Computing. Computing is a rapidly progressing field, opening new avenues and opportunities for growth and advancement. The model is fully cognizant with this fact, and it is oriented in a manner that it has capability to adapt the new changes that Computing brings about. In fact, this model will act as an agent of change itself as the research and development work is an integral part of it. The new trends are influenced by the industry and academia, but they will always fall into one of the components of the two-dimensional model of Computing educational infrastructure. There is room for further expansion in the model, but this expansion will take place within the components themselves. The components of the proposed model are discussed in the following section.

The said model is structured around the following dimensions:

1. ***Computing Academic Hub***
2. ***Computing Based Disciplines***

The following provides an overview of both dimensions:

**First Dimension: Computing Academic Hub**

The Computing Academic Hub is a philosophical representation of the root of the Computing framework from which the different focus areas branch out. The Computing academic hub defines the scope and the objectives to be attained from different Computing Educational Infrastructure components. The Computing academic hub will also probe the upheavals and new trends that emerge in its future. The graphical

model given in the following part of this Section demonstrates the structure of Computing Hub.

Computing is a professional discipline, accordingly, the interested students will undertake a respective undergraduate programme of a particular component as an entry to the profession of Computing. Therefore, it was imperative to structure graduate programme of a component of Computing on the basis of its respective undergraduate programme. The Committee invested substantial effort in designing undergraduate programmes that are more professional and thereby providing foundations for formal higher learning in the field.

The design of all three undergraduate programmes is two-dimensional. The first dimension pertains to the structure representing to the Computing requirements involving core, supporting areas and general education.

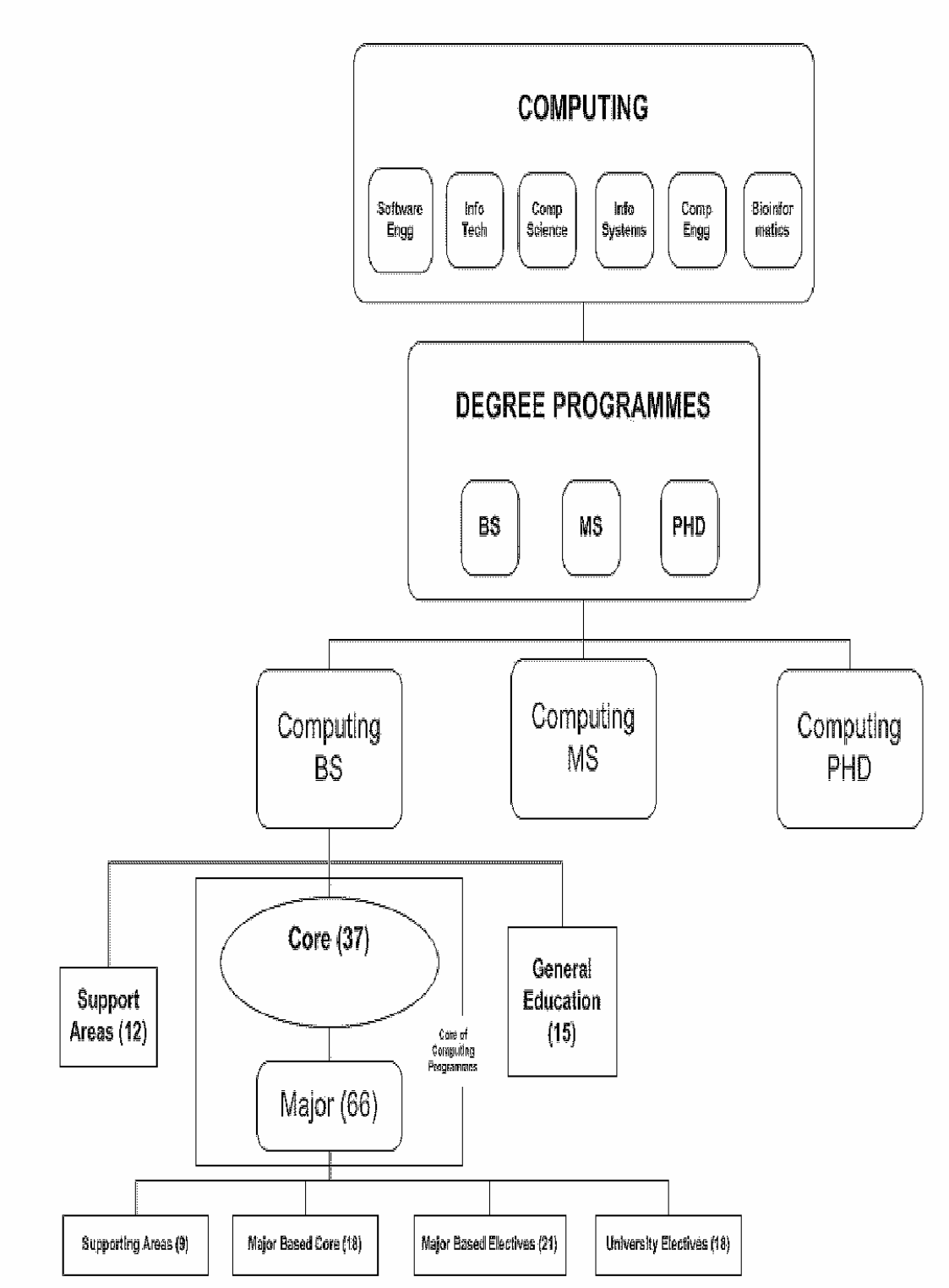
**Second Dimension: Computing Based Disciplines**

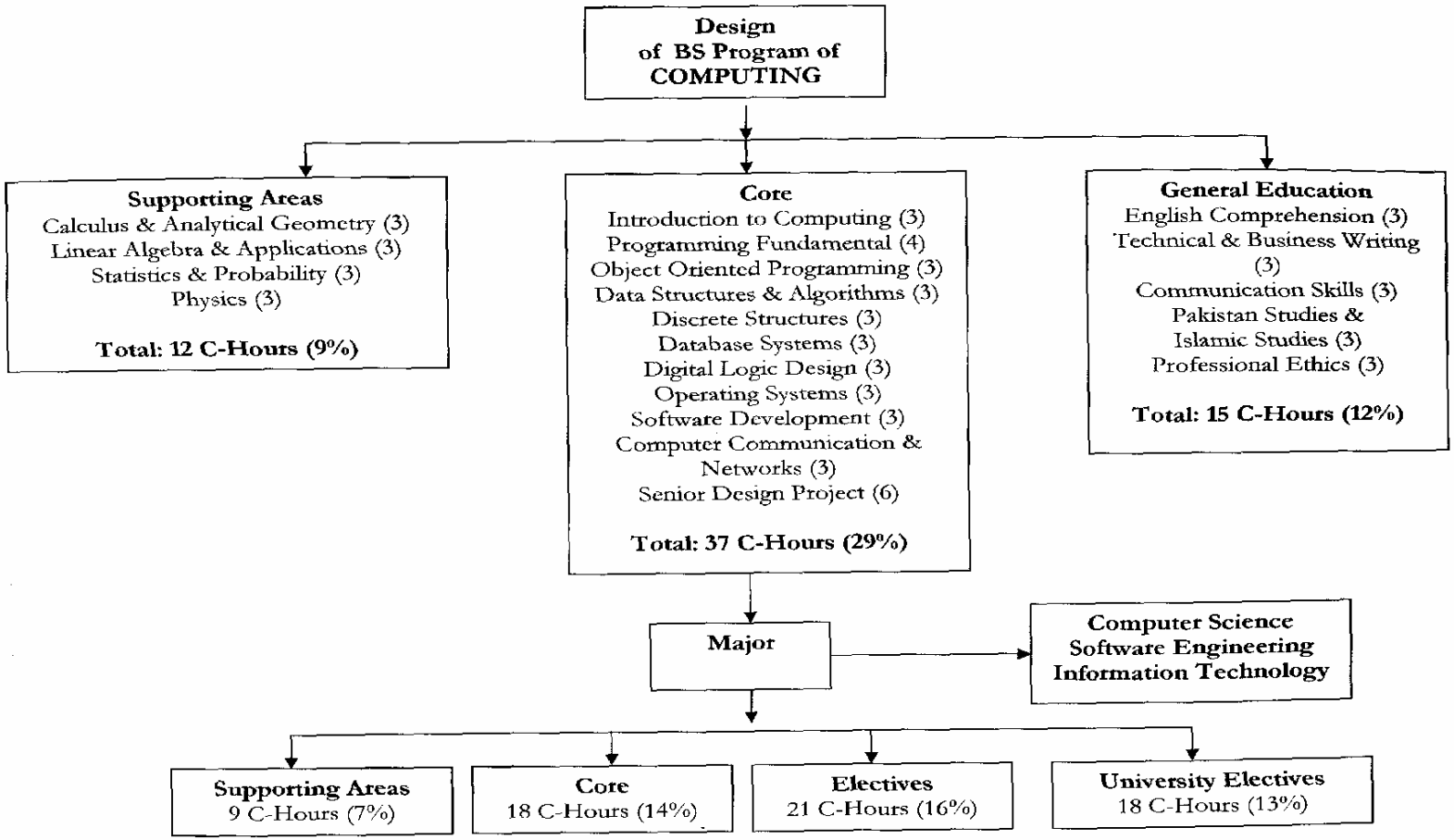
The second dimension of the model pertains to the family of disciplines of Computing. The design of two-dimensional model was envisioned to maintain the flexibility for continuous growth of Computing based disciplines. Essentially, the disciplines may be viewed an extension to the foundations of Computing in a particular branch of specialization. The structure of components involves domains including major based knowledge areas, supporting sciences and university based general education. The structure of major based knowledge area is further decomposed in to two tiers. The first is the major based core supplemented by the major based electives. Here, the second tier provides the requisite flexibility to maintain the currency with the latest development in the field in the given domain of the component. Moreover, a component of Computing should have its unique core to establish its identity in the large family of Computing.

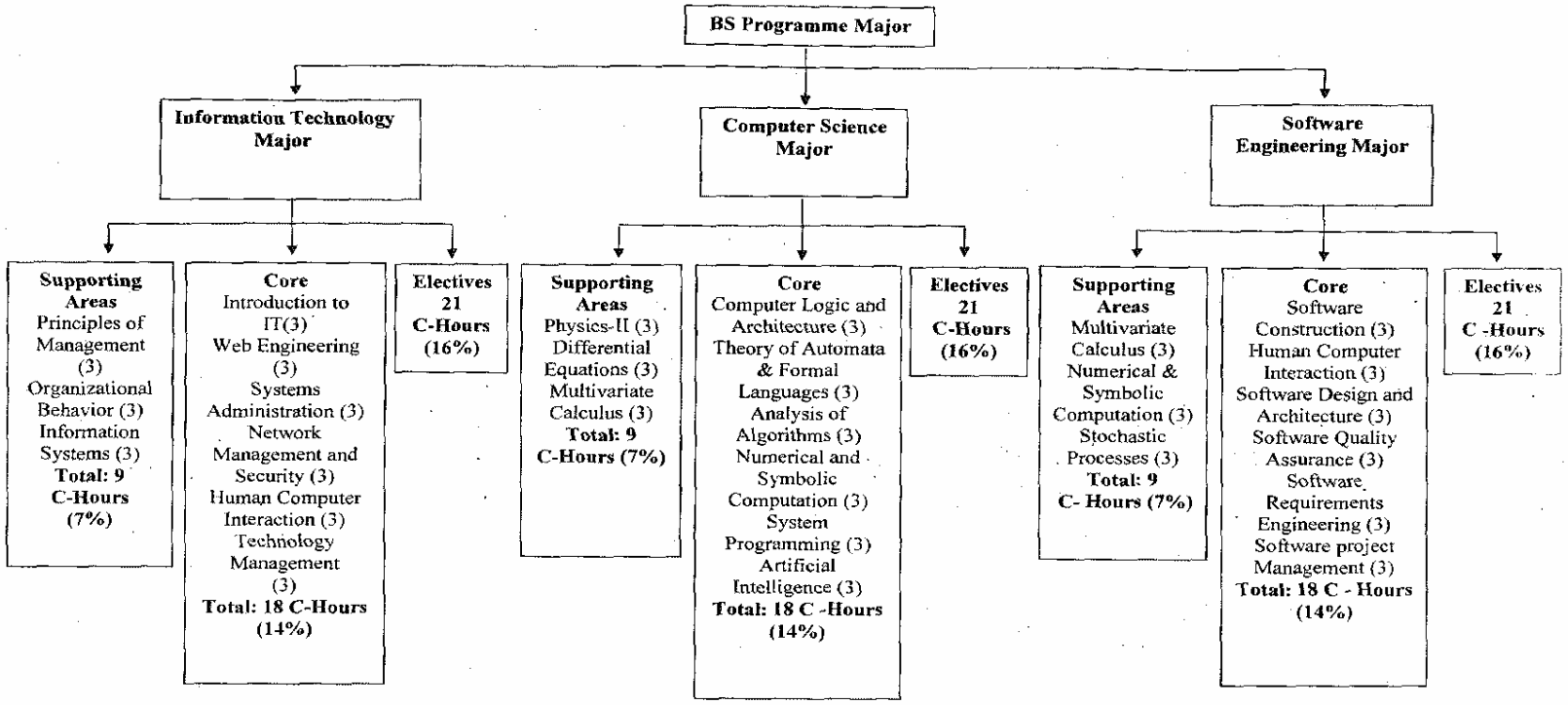
The complete curricula are documented regarding Computer Science, Software Engineering and Information Technology in the following pages and the overall model of the BS programme in Computing is presented in the following diagrams. All curricula were originally developed by their respective committees and finally reviewed by the Joint Committee.

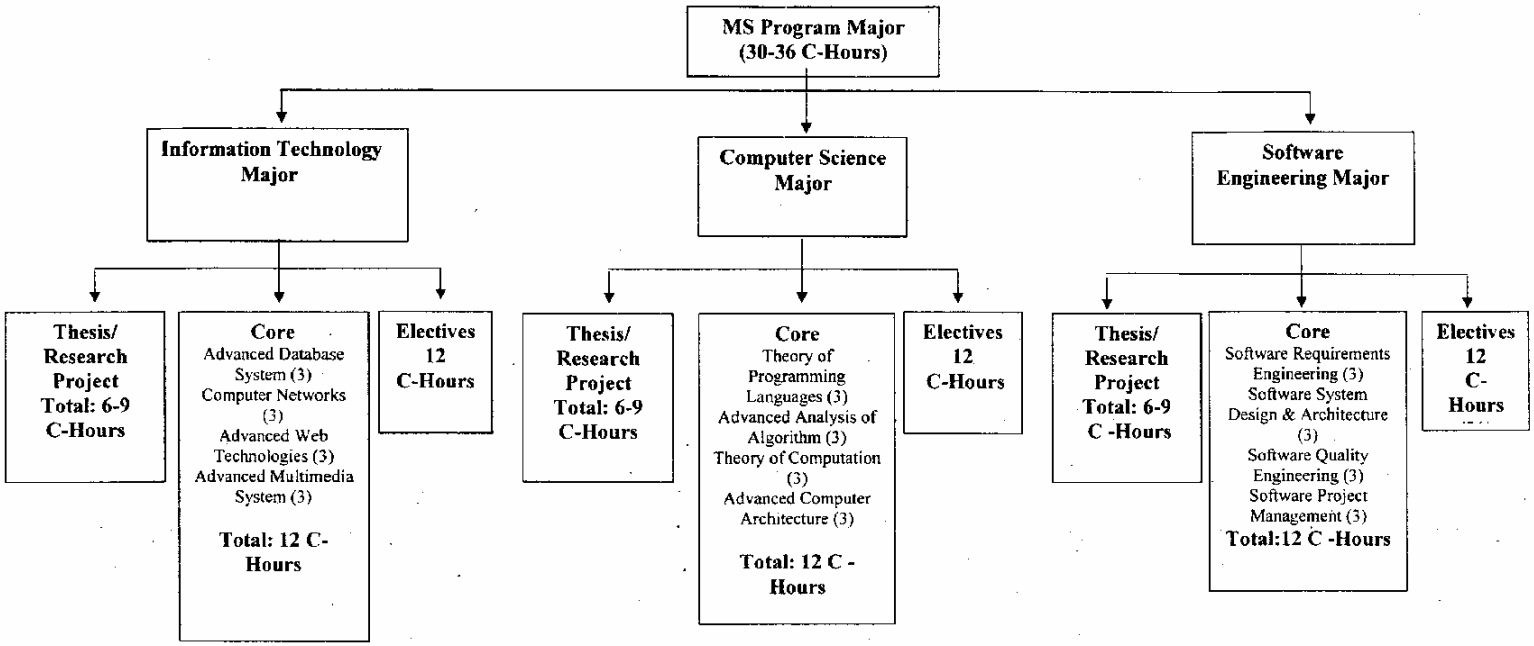
**Prof. Dr. Jamil Ahmad**
**Secretary**
**Dean, Iqra University**
**Islamabad**

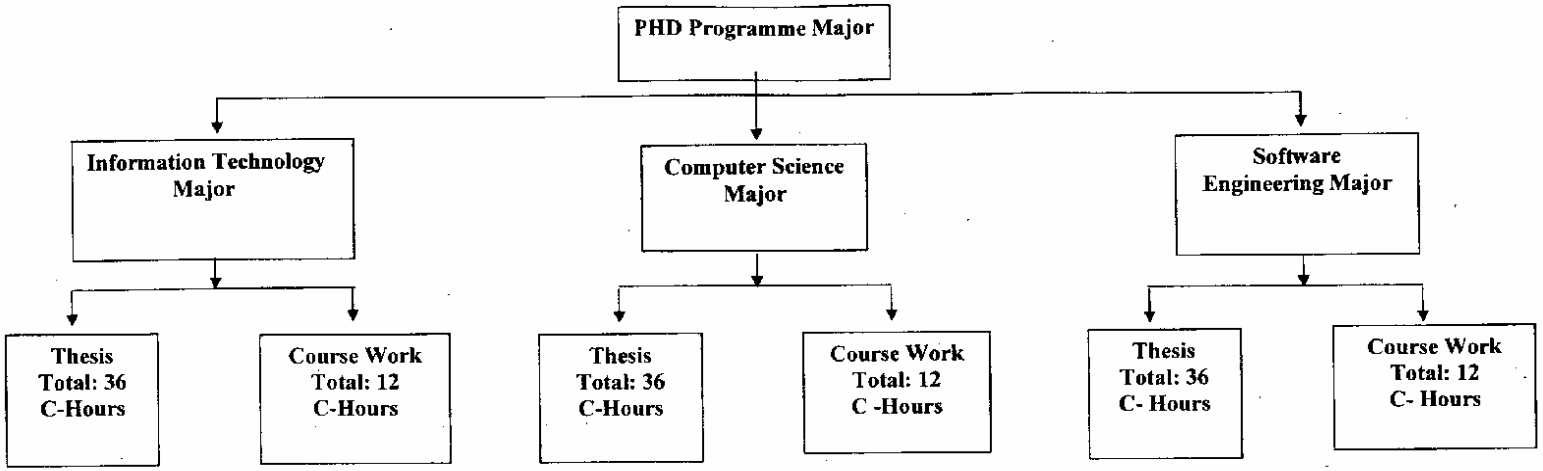
**Prof. Dr. Aftab Ahmad**
**Convener**
**Dean, Faculty of Engineering and Information Technology,**
**Foundation University Institute of Management and Computer Sciences, Rawalpindi**


COMPUTING
Software Engg
Info Tech
Comp Science
Info Systems
Comp Engg
Bioinfor matics
DEGREE PROGRAMMES
BS
MS
PHD
Computing BS
Computing MS
Computing PHD
Core (37)
Support Areas (12)
General Education (15)
Core of Computing Programmes
Major (66)
Supporting Areas (9)
Major Based Core (18)
Major Based Electives (21)
University Electives (18)


Design
of BS Program of
COMPUTING
Supporting Areas
Calculus & Analytical Geometry (3)
Linear Algebra & Applications (3)
Statistics & Probability (3)
Physics (3)
Total: 12 C-Hours (9%)
Core
Introduction to Computing (3)
Programming Fundamental (4)
Object Oriented Programming (3)
Data Structures & Algorithms (3)
Discrete Structures (3)
Database Systems (3)
Digital Logic Design (3)
Operating Systems (3)
Software Development (3)
Computer Communication & Networks (3)
Senior Design Project (6)
Total: 37 C-Hours (29%)
General Education
English Comprehension (3)
Technical & Business Writing (3)
Communication Skills (3)
Pakistan Studies &
Islamic Studies (3)
Professional Ethics (3)
Total: 15 C-Hours (12%)
Major
Computer Science
Software Engineering
Information Technology
Supporting Areas
9 C-Hours (7%)
Core
18 C-Hours (14%)
Electives
21 C-Hours (16%)
University Electives
18 C-Hours (13%)


BS Programme Major
Information Technology Major
Computer Science Major
Software Engineering Major
Supporting Areas
Principles of Management (3)
Organizational Behavior (3)
Information Systems (3)
Total: 9 C-Hours (7%)
Core
Introduction to IT(3)
Web Engineering (3)
Systems Administration (3)
Network Management and Security (3)
Human Computer Interaction (3)
Technology Management (3)
Total: 18 C-Hours (14%)
Electives
21 C-Hours (16%)
Supporting Areas
Physics-II (3)
Differential Equations (3)
Multivariate Calculus (3)
Total: 9 C-Hours (7%)
Core
Computer Logic and Architecture (3)
Theory of Automata & Formal Languages (3)
Analysis of Algorithms (3)
Numerical and Symbolic Computation (3)
System Programming (3)
Artificial Intelligence (3)
Total: 18 C -Hours (14%)
Electives
21 C-Hours (16%)
Supporting Areas
Multivariate Calculus (3)
Numerical & Symbolic Computation (3)
Stochastic Processes (3)
Total: 9 C- Hours (7%)
Core
Software Construction (3)
Human Computer Interaction (3)
Software Design and Architecture (3)
Software Quality Assurance (3)
Software Requirements Engineering (3)
Software project Management (3)
Total: 18 C - Hours (14%)
Electives
21 C -Hours (16%)


MS Program Major
(30-36 C-Hours)
Information Technology Major
Computer Science Major
Software Engineering Major
Thesis/ Research Project Total: 6-9 C-Hours
Core
Advanced Database System (3)
Computer Networks (3)
Advanced Web Technologies (3)
Advanced Multimedia System (3)
Total: 12 C-Hours
Electives 12 C-Hours
Thesis/ Research Project Total: 6-9 C-Hours
Core
Theory of Programming Languages (3)
Advanced Analysis of Algorithm (3)
Theory of Computation (3)
Advanced Computer Architecture (3)
Total: 12 C - Hours
Electives 12 C-Hours
Thesis/ Research Project Total: 6-9 C -Hours
Core
Software Requirements Engineering (3)
Software System Design & Architecture (3)
Software Quality Engineering (3)
Software Project Management (3)
Total:12 C -Hours
Electives 12 C-Hours


PHD Programme Major
Information Technology Major
Computer Science Major
Software Engineering Major
Thesis Total: 36 C-Hours
Course Work Total: 12 C-Hours
Thesis Total: 36 C-Hours
Course Work Total: 12 C -Hours
Thesis Total: 36 C- Hours
Course Work Total: 12 C- Hours

***Computing***

# Scheme of Studies for Bachelor Degree Programmes

***(Computer Science, Software Engineering, Information Technology)***

## Computing — Requirements for Bachelor Degree Programmes

| Required Computing Courses | | |
|---|---|---|
| # | **Knowledge Area** | **Credit hours** |
| 1 | Computing Core Areas | 37 |
| 2 | Supporting Sciences | 12 |
| 3 | General Education | 15 |
| **Total** | | **64/130** |

## Computing — Core Courses (37 Credits Hours)

| Required Computing Courses | | | | | |
|---|---|---|---|---|---|
| # | **Code** | **Preq** | **Course Title** | **Credit hours** | **Proposed Semester** |
| 1 | CS | - | Introduction to Computing | 3 (2-0) | 1 |
| 2 | CS | - | Programming Fundamentals | 4 (3-3) | 1 |
| 3 | CS | 2 | Object Oriented Paradigm | 3 (3-3) | 2 |
| 4 | CS | - | Discrete Structures | 3 (3-0) | 2 |
| 5 | CS | 3 | Data Structure and Algorithms | 3 (3-3) | 3 |
| 6 | CS | 1, 4 | Digital Logic and Computer Architecture | 3 (2-3) | 3 |
| 7 | CS | 5 | Operating Systems | 3 (2-3) | 4 |
| 8 | CS | 5 | Database Systems | 3 (3-3) | 4 |
| 9 | CS | 5 | Introduction to Software Development | 3 (3-3) | 5 |
| 10 | CS | 7 | Computer Communications and Networks | 3 (2-3) | 6 |
| 11 | CS | - | Senior Design Project **(37/130)** | 6 (0-18) | 7, 8 |

## Computing — Supporting Sciences (12 Credits Hours)

| Required Supporting Courses | | | | | |
|---|---|---|---|---|---|
| # | **Code** | **Preq** | **Course Title** | **Credit hours** | **Proposed Semester** |
| 12 | MT | - | Calculus and Analytical Geometry | 3 (3-0) | 1 |
| 13 | MT | - | Probability and Statistics | 3 (3-0) | 2 |
| 14 | MT | - | Linear Algebra | 3 (3-0) | 4 |
| 15 | PH | - | Physics (Electromagnetism) **(12/130)** | 3 (3-0) | 3 |

**Computing — General Education (15 Credits Hours)**

| Required General Education Courses | | | | | |
|---|---|---|---|---|---|
| # | **Code** | **Preq** | **Course Title** | **Credit hours** | **Proposed Semester** |
| 1 | EG | - | English Composition and Comprehension | 3 (3-0) | 1 |
| 2 | EG | - | Technical and Business Writing | 3 (3-0) | 2 |
| 3 | EG | - | Communication Skills | 3 (3-0) | 3 |
| 4 | PK | - | Islamic and Pakistan Studies | 3 (3-0) | 1 |
| 5 | SS | - | Professional Practices **(15/130)** | 3 (3-0) | 8 |

# Computing — Detail of Courses

## Core Courses (37 Credits Hours)

| | |
|---|---|
| **Course Name:** Introduction to Computing | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **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:**<br>Computers: Information Technology in Perspective, 9/e by Larry Long and Nancy Long,<br>Prentice Hall, 2002/ISBN: 0130929891.<br>*An Invitation to Computer Science*, Schneider and Gersting, Brooks/Cole Thomson Learning, 2000.<br>*Computer Science: An overview of Computer Science*, Sherer. | |

| | |
|---|---|
| **Course Name:** Programming Fundamentals | |
| **Course Structure:** Lectures: 3 / Labs: 3 | **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:** 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:**<br>*Programme Design with Pseudo-code*, Bailey and Lundgaard, Brooks/Cole Publishing, 1988<br>*Simple Programme Design: A step-by-step approach*, 4/e, Lesley Anne Robertson, ISBN: 0-619-16046-2 © 2004. | |

| | |
|---|---|
| **Course Name:** Object Oriented Paradigms | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Introduction to Computing, Programming Fundamentals | |
| **Objectives:** The course aims to focus on object-oriented concepts, analysis and software development. | |
| **Course Outline:** Evolution of OO, 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:**<br>*Understanding Object Oriented Programming*, Budd, Addison Wesley.<br>*Java: How to Programme*, 5/e, Deitel and Deitel, Prentice Hall, 0131016210/ 0131202367 International Edition.<br>*C++: How to Programme*, Deitel and Deitel, 4/e, Pearson.<br>*Thinking in C++*, 2nd Edition, Bruce Eckel, Prentice Hall. | |

| | |
|---|---|
| **Course Name:** Data Structures and Algorithms | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Object Oriented Paradigms | |
| **Objectives:** The course is designed to teach students structures and schemes, which allow them to write programmes to efficiently manipulate, store, and retrieve data. Students are exposed to the concepts of time and space complexity of computer programmes. | |
| **Course Outline:** Introduction to data structures; Arrays, Stacks, Queues, Priority Queues, Linked Lists, Trees, and Graphs. Recursion, sorting and searching algorithms, Hashing, Storage and retrieval properties and techniques for the various data structures. Algorithm Complexity, Polynomial and Intractable Algorithms, Classes of Efficient Algorithms, Divide and Conquer, Dynamic, Greedy | |
| **Reference Material:**<br>*Data Abstraction and Problem Solving with C++, 2nd ed, Frank M. Carrano, Paul Helman, Robert Veroff, Addison-Wesley, 1998.*<br>*Data Structures and Algorithms* (SAMS teach yourself), Lafore, Sams Publishing, 1999.<br>*Fundamentals of Data Structures in C++*, Horowitz, Sahni, and Mehta, Computer Science Press, 1995.<br>*Data Structures in JAVA*, Standish, Addison Wesley, 2000 | |

| | |
|---|---|
| **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.

**Course Outline:** Introduction to logic and proofs: Direct proofs; proof by contradiction, Sets, Combinatorics, Sequences, Formal logic, Prepositional and predicate calculus, Methods of Proof, Mathematical Induction and Recursion, loop invariants, Relations and functions, Pigeon whole principle, Trees and Graphs, Elementary number theory, Optimization and matching. Fundamental structures: Functions (surjections, injections, inverses, composition); relations (reflexivity, symmetry, transitivity, equivalence relations); sets (Venn diagrams, complements, Cartesian products, power sets); pigeonhole principle; cardinality and countability.

**Reference Material:**
*Discrete Mathematics and Its Applications,* 5th edition; by Rosen; McGraw-Hill; 0-07-242434-6.
*Discrete Mathematics by Richard Johnsonbaugh,* Prentice Hall, 0135182425.
Discrete Mathematical Structures, 4th Edition, by Kolman, Busby & Ross, 2000, Prentice-Hall, ISBN: 0-13-083143-3.

| | |
|---|---|
| **Course Name:** Digital Logic and Computer Architecture | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours: 3** |
| **Prerequisites:** Discrete Structures, Introduction to Computing | |
| **Objectives:** This course introduces the concept of digital logic, gates and the digital circuits. Further, it focuses on the design and analysis combinational and sequential circuits. It also serves to familiarize the student with the logic design of basic computer hardware components. | |

**Course Outline:** Overview of Binary Numbers, Boolean Algebra, switching algebra, and logic gates, Karnaugh Map and Quin-McCluskey methods, simplification of Boolean functions, Combinational Design; two level NAND/NOR implementation, Tabular Minimization, Combinational Logic Design: adders, subtracters, code converters, parity checkers, multilevel NAND/NOR/XOR circuits, MSI Components, design and use of encoders, decoders, multiplexers, BCD adders, and comparators, Latches and flip-flops, Synchronous sequential circuit design and analysis, Registers, synchronous and asynchronous counters, and memories, Control Logic Design, Wired logic and characteristics of logic gate families, ROMs, PLDs, and PLAs, State Reduction and good State Variable Assignments, Algorithmic State Machine (ASM) Charts, Asynchronous circuits, Memory systems, Functional organization, Multiprocessor and alternative architectures: Introduction to SIMD, MIMD, VLIW, EPIC; systolic architecture; interconnection networks; shared memory systems; cache coherence; memory models and memory consistency, Performance enhancements, Contemporary architectures.

**Reference Material:**
*Digital Design*, 2nd Ed., M. Morris Mano, Prentice Hall, 1991.
*Practical Digital Logic Design and Testing*, P K Lala, Prentice Hall, 1996.

**Course Name:** 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 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 recovery techniques; Database security and authorization. Small Group Project implementing a database. Physical database design: Storage and file structure; indexed files; hashed files; signature files; b-trees; files with dense index; files with variable length records; database efficiency and tuning Data Warehousing and Data Mining, Emerging Database Technologies and Applications.

**Reference Material:**
*Database Systems,* C.J.Date, Addison Wesley Pub. Co. (2004).
*Database Systems: A Practical Approach to Design, Implementation and Management,* R.Connolly and P.Begg, Addison-Wesley Pub. Co (2003).
Fundamentals of Database Systems, 3/E, Elmasri and Navathe, Addison-Wesley, ISBN: 0-201-74153-9.

**Course Name:** Operating Systems

| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
|---|---|
| **Prerequisites:** Data Structures and Algorithms | |
| **Objectives:** The objective of this course is to give students knowledge of construction and working of Operating systems, to enable them to understand management and sharing of computer resources, communication and concurrency and develop effective and efficient applications and also to appreciate the problems and issues regarding multi-user, multitasking, and distributed systems. | |
| **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.<br><br>Lab assignments involving different single and multithreaded OS algorithms. | |
| **Reference Material:**<br>*Applied Operating Systems Concepts,* 6th Edition, Silberschatz A., Peterson, J.L., & Galvin P.C. 1998.<br>*Modern Operating Systems,* 2nd Edition, Tanenmaum A.S., 2001. | |

| **Course Name:** Computer Communication and Networks | |
|---|---|
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Operating Systems | |
| **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 or projects involving implementation of protocols at different layers. | |
| **Reference Material:**<br>Introduction to Computer Networks, Tanenbaum<br>Unix Network Programming, Richard Stevens<br>*Computer networks: a systems approach*, Larry Peterson, Bruce Davie, Princeton Univ., Princeton.<br>Computer Networking: A Top-Down Approach Featuring the Internet, 2/e, James F Kurose, Keith W Ross, Addison Wesley 2003. ISBN: 0-201-97699-4. | |

| | |
|---|---|
| **Course Name:** Introduction to Software Development | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours: 3** |
| **Prerequisites:** Data Structures and Algorithms | |
| **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:**<br>*Software Engineering: A Practioner's Approach,* Roger Pressman, McGraw-Hill, 2001.<br>*Object-Oriented Software Engineering,* Stephan Schach, Irwin, 1999. | |

| | |
|---|---|
| **Course Name:** Senior Software Project | |
| **Course Structure:** Lectures: 0 / Labs: 18 | **Credit Hours:** 6 |
| **Prerequisites:** Software Engineering – I, 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:**<br>*Software Project Management in Practice* by Jalote, Pankaj. | |

## Computing-Supporting Sciences (12 Credits Hours)

| | |
|---|---|
| **Course Name:** Calculus and Analytic Geometry | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** To build the basic 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:**<br>Calculus and Analytical Geometry By Swokowski, Olinick and Pence.<br>Calculus*, H. Anton, John Wiley and Sons (WIE), ISBN: 0471572608*.<br>Calculus, William E. Boyce Richard C. Diprima, John Wiley & Sons, ISBN: 0471093335 | |

| | |
|---|---|
| **Course Name:** Probability and Statistics | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** | |
| **Objectives:** To introduce the concept of statistics, randomness and probability and build on these concepts to develop tools and techniques to work with random variables | |
| **Course Outline:** Introduction to Statistics, Descriptive Statistics, Statistics in decision making, Graphical representation of Data Stem-and Lead plot, Box-Cox plots, Histograms and Ogive, measures of central tendencies, dispersion for grouped and ungrouped Data, Moments of frequency distribution; examples with real life, use of Elementary statistical packages for explanatory Data analysis. Counting techniques, definition of probability with classical and relative frequency and subjective approaches, sample space, events, laws of probability. *General Probability Distributions,* Conditional probability and Bayes theorem with application to Random variable (Discrete and continuous) Binomial, Poisson, Geometric, Negative Binomial Distributions; Exponential Gamma and Normal distributions. *Regression and Correlation.* | |
| **Reference Material:**<br>*Introduction to Statistics,* Walpole, 1982 Prentice Hall, ISBN: 0024241504.<br>*Statistical Data Analysis*, G. Cowan G, 1998, Clarendon, Oxford.<br>*Advances in Statistical Analysis and Statistical Computing III* Mariano R (Ed.), (1993), JAI Press, Greenwich, Conn. | |

| | |
|---|---|
| **Course Name:** Physics (Electromagnetism) | |
| **Course Structure:** Lectures: 2 Labs: 2 | **Credit Hours:** 3 (2+1) |
| **Prerequisites:** None | |
| **Course Outline:** Review of Vectors, *Electric Charge:* Coulomb's Law, electric field and intensity, electric potential, capacitors and charge storage concepts, *Magnetism:* magnetic fields, Faraday's and Lenz's Laws, Ampere's law and its applications, Eddy Currents, inductance, induced current and their applications, definitions of the values of AC signals (Average and RMS Values), *Electric and Magnetic circuits:* Electric current, resistance, Ohm's Law, simple resistive circuits (series and parallel), Kirchoff laws, Network theorems (DC analysis), RC and RL circuits. *Fundamentals of Semiconductor physics:* Band theory, semiconductors (intrinsic and extrinsic), pn junction, pn- junctions as a rectifier. | |
| **Reference Material:**<br>*University Physics* by Freedman and Young (10 th and higher editions),<br>*College Physics* by Resnick, Halliday and Krane (6th and higher edition) | |

| | |
|---|---|
| **Course Name:** Linear Algebra | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Objectives:** Fundamentals of Linear Algebra with emphasis on problem-solving | |
| **Course Outline:** Vectors, Vector Spaces, Matrices & Determinants, Cofactor and Inverse, Rank, Linear Independence, Solution of Linear systems, Gaussian Elimination, Positive Definite matrix, Linear Transformations, Operations on matrices, Inner products, Eigenvalues & Eigenvectors. Applications to Systems of Equations and to Geometry. | |
| **Reference Material:**<br>*Linear Algebra,* David C Lay, Pearson Addison Wesley, 1999, ISBN: 0201660369 | |

## Computing-General Education (15 Credits Hours)

| | |
|---|---|
| **Course Name:** English Composition and Comprehension | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** To develop good English writing, language usage and reading skills. | |
| **Course Outline:** Principles of writing good English, understanding the composition process: writing clearly; word, sentence and paragraph. Comprehension and expression. Use of grammar and punctuation. Process of writing, observing, audience analysis, collecting, composing, drafting and revising, persuasive writing, reading skills, listening skills and comprehension, skills for taking notes in class, skills for exams. | |
| **Reference Material:** *Warriner's English Grammar and Composition*, John E. Warriner | |

| | |
|---|---|
| **Course Name:** Communication Skills | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |

| |
|---|
| **Objectives:** To develop good English writing, language usage and reading skills. To appreciate the importance of business communication and to develop understanding of communication concepts, principles, theories and problems. To develop good oral communication and presentation skills. |
| **Course Outline:** Principles of writing good English, understanding the composition process: writing clearly; words, sentence and paragraphs. Comprehension and expression. Use of grammar and punctuation. Process of writing, observing, audience collecting, composing, drafting and revising, persuasive writing, reading skills, listening skills and comprehension, skills for taking notes in class, skills for exams. Business communications; planning messages, writing concise but with impact. Letter formats, mechanics of business, letter writing, letters, memo and applications, summaries, proposals, writing resumes, styles and formats, oral communications, verbal and non-verbal communication, conducting meetings, small group communication, taking minutes. Presentation skills; presentation strategies, defining the objective, scope and audience of the presentation, material gathering material organization strategies, time management, opening and concluding, use of audio-visual aids, delivery and presentation. |
| **Reference Material:**<br>*Business English,* Vawdrey, Stoddard, Bell. |

| | |
|---|---|
| **Course Name:** Technical and Business Writing | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** To develop efficient literature survey, analysis, report writing and document designing skills. | |
| **Course Outline:** Overview of technical reporting, use of library and information gathering, administering questionnaires, reviewing the gathered information. Technical exposition; topical arrangement, exemplification, definition, classification and division, casual analysis, effective exposition, technical narration, description and argumentation, persuasive strategy. Organizing information and generation solution: brainstorming, organizing material, construction of the formal outline, outlining conventions, electronic communication, generation solutions. Polishing style: paragraphs, listening sentence structure, clarity, length and order, pomposity, empty words, pompous vocabulary, document design: document structure, preamble, summaries, abstracts, table of contents, footnotes, glossaries, cross-referencing, plagiarism, citation and bibliography, glossaries, index, appendices, typesetting systems, creating the professional report; elements, mechanical elements and graphical elements. Reports: Proposals, progress reports, Leaflets, brochures, handbooks, magazines articles, research papers, feasibility reports, project reports, technical research reports, manuals and documentation, thesis. Electronic documents, Linear verses hierarchical structure documents. | |
| **Reference Material:**<br>Greenfield, T., Research Methods, Guidance for Postgraduates, Arnold, 1996, 034064629. | |

| | |
|---|---|
| **Course Name:** Islamic & Pakistan Studies | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** To impart an understanding of the fundamental principles and teachings of Islam through study of selected verses of the Quran and Prophetic Sayings. Important facets of the Prophet's life and salient, features of Islamic Civilization. To provide appreciation of other prominent religions, systems of ethics and cultures to prepare students to survive in international and multicultural work place.<br><br>To take an analytical view in the history and development of Muslim society and culture in the sub-continent, emergence of Pakistan and its constitutional development. To develop an appreciation of the issues and challenges currently being faced in Pakistan. The strengths of its people and strategies to deal with the impediments to progress. International relations of Pakistan | |
| **Course Outline:** Fundamentals of Islam. (Aqaid, Ibadat, Islamic Dawah etc.); Ethical values of Islam; Ser ah of the Holy Prophet (PBUH); Islamic Civilization and its affects on humanity. Study of other prominent world religions and ethical systems in comparison with Islamic viewpoint. Multicultural societies.<br><br>Historical background of Pakistan: Muslim society in Indo-Pakistan, the movement led by the societies, The downfall of Islamic society, The establishment of British Raj- Causes and consequences. Political evolution of Muslims in the twentieth century: Sir Syed Ahmed Khan; Muslim League; Nehru; Allama Iqbal: Independence Movement; Lahore Resolution; Pakistan culture and society, Constitutional and Administrative issues, Pakistan and its geo-political dimension, Pakistan and International Affairs, Pakistan and the challenges ahead. | |
| **Reference Material:**<br>*Chaudary M. Ali ,The Emergence of Pakistan, 1967.*<br>*K.K.Aziz,The making of Pakistan, 1976.* | |

| | |
|---|---|
| **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. | |

| |
|---|
| **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:**<br>*Professional Issues in Software Engineering*, M.F. Bott et al. |

# Computer Science Curricula 2004

## National Curriculum Revision Committee – Computer Science

A three days final meeting of the National Curriculum Revision Committee (Computer Science) pertaining to the development of curricula for Computer Sciences degree programmes and deliberation of related matters was held on March 15-17, 2004 at Higher Education Commission (HEC), Islamabad. The primary objective of the meeting was to discuss and finalize the curricula drafted by the same committee in the last meeting held on December 15-17, 2003. Following attended the meeting:

1. Dr. Aftab Ahmad **Convener**
   Professor and Dean
   Faculty of Engineering & Information Technology
   Foundation University Institute of Management &
   Computer Sciences
   Rawalpindi

2. Dr. Jamil Ahmad **Secretary**
   Professor and Dean
   Iqra University
   Islamabad Campus

3. Dr. N. A. Sangi **Member**
   Professor and Dean
   Faculty of Science
   Allama Iqbal Open University
   Islamabad

4. Dr. Khalid Rashid **Member**
   Professor and Dean
   Faculty of Management Sciences and Applied Sciences
   International Islamic University
   Islamabad

5. Prof. Dr. Ayub Alvi **Member**
   Dean, Faculty of Computer Sciences
   FAST-National University of Computer &
   Emerging Sciences
   Islamabad

6. Dr. Mohammad Riaz **Member**
   Professor and Dean
   Faculty of Computer Sciences
   Bahria University
   Islamabad

7. Dr. Mohammad Jaffer-ur-Rehman **Member**
   Professor and Dean
   Faculty of Engineering & Computer Sciences
   Mohammad Ali Jinnah University
   Islamabad

8. Dr. Syed Mansoor Sarwar **Member**
   Professor
   Department of Computer Science
   LUMS
   Lahore

9. Dr. Romana Aziz **Member**
   Associate Professor
   Foundation University Institute of
   Management Sciences
   Rawalpindi

10. Dr. Naveed Ikram **Member**
    Director
    Riphah Institute of Informatics
    Islamabad

11. Dr. M. Afzal Bhatti **Member**
    Professor
    Department of Computer Science
    Quaid-i-Azam University
    Islamabad

12. Mr. Irfan Ahmad **Member**
    Ph. D Programme Coordinator
    Punjab University College of IT
    Lahore

13. Dr. Iftikhar Hussain Shah **Member**
    Professor and Chairman
    Department of Computer Sciences
    Government College University
    Lahore

14. Mr. Abdul Aziz Sabir **Member**
    Associate Professor and Chairman
    Department of Computer Sciences
    University of Agriculture
    Faisalabad

15. Mr. Mohammad Shahid **Member**
    Assistant Professor
    Institute of Business Management &
    Computer Sciences
    NWFP Agricultural University
    Peshawar

16. Mr. Abdul Mohsin **Member**
    Lecturer
    Iqra University
    Quetta

17. Prof. Dr. Mohammad Salim Chandio **Member**
    Institute of Maths & Computer Sciences
    University of Sindh
    Jamshoro

18. Mr. Mohammad Zahid **Member**
    Associate Professor and Chairman
    Department of Computer System &
    Software Engineering
    Mehran University of Engineering & Technology
    Jamshoro

19. Dr. Jawad Qamar **Member**
    Professor and Dean
    Faculty of Computer Sciences & Informatics
    Dadabhoy Institute of Higher Education
    Karachi

20. Dr. Aftab A. Maroof **Member**
    Professor and Director
    FAST-NU
    Islamabad

21. Mr. Mohammad Daud Khattak **Member**
    Assistant Professor
    Multimedia Electronic Courseware Design Centre
    Department of Computer Sciences
    Allama Iqbal Open University
    Islamabad

22. Dr. Jamil Sawar **Member**
    Director
    Barani Institute of IT
    Rawalpindi

23. Dr. Massod A. Malik — **Member**
    Professor
    Barani Institute of IT
    Rawalpindi

24. Mr. Nasir Uddin Umar — **Member**
    Assistant Professor
    Federal Urdu University
    Islamabad

25. Mr. Zafar Malik — **Member**
    Associate Dean
    Muhammad Ali Jinnah University
    Islamabad

26. Mr. Khalid Mengal — **Member**
    Lecturer
    Faculty of Engineering Sciences
    Balochistan University of Information Technology
    and Management Sciences,
    Quetta

27. Mr. Tariq Jameel — **Special Invitation**
    Chairman
    Techno-ed (Pvt) Ltd

28. Maj. (Retd.) Ifthikhar Naqwi — **Special Invitation**
    Techno-ed (Pvt) Ltd.

The following members could not attend the meeting:

1. Prof. Dr. Shaiq A. Haq — **Member**
   Chief Consultant
   IT Research
   University of Engineering & Technology
   Lahore

2. Dr. Altaf H. Khan — **Member**
   Professor & Dean
   Institute of Management & Technology
   Lahore

3. Mr. Dost Mohammad Khan — **Member**
   Incharge
   Department of Computer Sciences
   Islamia University
   Bahawalpur

4. Prof. Dr. Mumtaz Hussain Mahar **Member**
   Chairman
   Department of Computer Sciences
   Shah Abdul Latif University
   Khairpur

5. Prof. Dr. Salim-ur-Rehman **Member**
   Sarhad University of Science & Technology
   Peshawar

6. Prof. Dr. Mehboob Yaseen **Member**
   Dean
   Faculty of Computer Sciences
   GIK
   Swabi

7. Dr. Athar Mahboob **Member**
   Assistant Professor
   IBA
   Karachi

8. Mr. Ikram ul Haq **Member**
   Associate Professor
   COMSAT Institute of Information Technology
   Islamabad

9. Dr. Shoab Ahmad Khan **Member**
   Associate Professor
   Department of Computer Engineering
   College of E&ME
   NUST
   Rawalpindi

10. Mr. Muhammad Naeem **Member**
    Assistant Professor
    Department of Computer Science
    University of Peshawar
    Peshawar

The Meeting started with recitation from the Holy Quran and a welcome address by Professor Dr. Aftab Ahmad, Convener, National Curriculum Revision Committee (NCRC). He informed the participants that the comprehensive report of previous meeting held on December 15-17, 2003 was circulated among all members and would be confirmed in this meeting. He also requested the participants to setup an agenda of items of discussion for the three days meeting in order to achieve the desired objectives of the Committee. The following programmes were discussed by the participants:

1. BS (CS) Programme
2. MS (CS) Programme
3. Design of Ph.D. (CS) Programme

Moreover, Prof. Dr. Aftab Ahmad, Convener, nominated Dr. Jamil Ahmad, Dean, Iqra University Campus, Islamabad for the responsibility of Secretary of National Curriculum Revision Committee due to absence of Mr. Daud Khattak. The nomination was unanimously agreed by all members of the Committee.

## Curriculum for BS (Computer Science) Programme:

Most of the participants took part in the general discussion on this item of the agenda. It was discussed thoroughly using the following basis suggested in the previous meeting by the Convener of the Committee:

- **Objectives/Goals**
- **Strategies**
- **Fast Changing Disciplines**
- **Emerging Technologies**
- **International Standards**
- **Industrial Challenges**
- **Possible Programme Design Structures**

The BS programme, proposed in the last meeting, was discussed thoroughly and compared its structure with the recommendations of various international bodies including IEEE and ACM. The report entitled **"Curricula 2004 of ACM and IEEE Curriculum Task Force"** was mainly discussed in detail. Some changes were recommended in all sections of the curricula developed by the Committee in the last meeting. The Committee finally agreed to the curriculum model presented in the following table.

| # | Category | | Credit Hrs |
|---|---|---|---|
| 1 | **Computing courses** | | |
| | Computing – Core courses | 37 | |
| | Computing - Supporting areas | 12 | |
| | Computing - General Education | 15 | 64 |
| 2 | **Computer Science courses** | | |
| | CS – Core courses | 18 | |
| | CS – Electives | 21 | |
| | CS – Supporting Area | 9 | 48 |
| 3 | **University Electives** | | 18 |
| | **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:

- The programme should provide a broad understanding of the field via introducing concepts, theory, and techniques.
- Intensive education/training in focused areas of Computer Science is desirable.
- The programme may encourage students to develop and use abstract models in addition to apply respective technology in practical situations.
- 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.
- The programme should provide formal foundations for higher learning.
- The programme should be dynamic and flexible enough to maintain currency with the latest scientific and technological developments in the field.
- The programme should provide professional orientation to prepare students for industry.

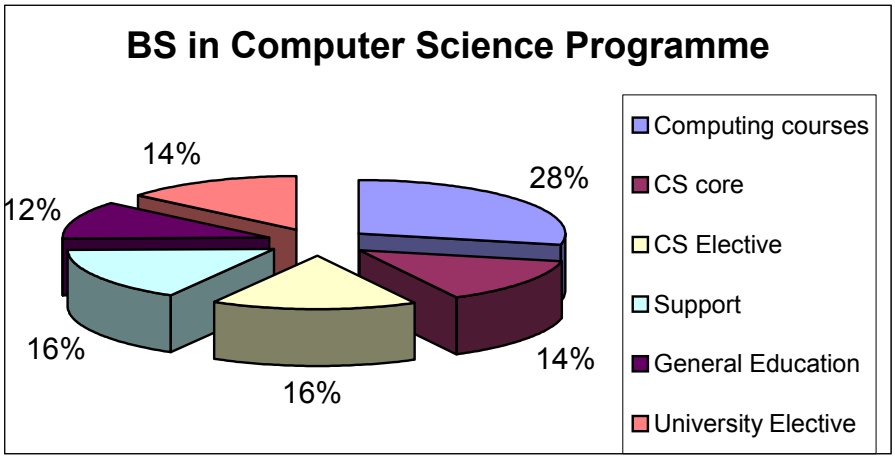
**Programme Structure**

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 130 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.




| Type | Computing | | Computer Science | | Support | | General | |
|---|---|---|---|---|---|---|---|---|
| | Cr. Hours | %age | Cr. Hours | %age | Cr. Hours | %age | Cr. Hours | %age |
| **Core** | 37 | 28% | 18 | 14% | 21 | 16% | 15 | 12% |
| **Electives** | - | - | 21 | 16% | - | - | 18 | 14% |
| Total | 37 | 28% | 39 | 30% | 21 | 16% | 33 | 26% |

- Some clusters regarding Computer Science Electives are listed below:

I. Networking
II. Database
III. Intelligent Systems
IV. Graphics & Visualization
V. Software Engineering
VI. Web Engineering
VII. E-Commerce
VIII. Multimedia
IX. Distributed Computing
X. Security

**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 sate 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:

- The IEEE (Institute of Electronic and Electrical Engineers) Software Engineering Proposal
- The IFIP (International Federation of Information Processing) Curriculum Reports
- The DPMA (Data Processing Management Association) Computer Systems Proposal
- The ACM (Association of Computing Machinery) Curriculum Task Force-Curriculum 2001
- The ITAA (Information Technology Association of America) Report on IT Workforce Study

## Courses for BS programme:

**Bachelor of Science in Computer Science; BS (CS)**
**Computing Core Requirements 37 Credit Hours (Refer to Computing part)**

| Required Computer Science Courses | | | | | |
|---|---|---|---|---|---|
| # | Code | Preq | Course Title | Credit hours | Proposed Semester |
| 12 | CS | 2 | Computer Organization and Assembly Language | 3 (3-0) | 4 |
| 13 | CS | 4 | Theory of Automata & Formal Languages | 3 (3-0) | 6 |
| 14 | CS | 5 | Analysis of Algorithms | 3 (3-0) | 6 |
| 15 | CS | - | Artificial Intelligence | 3 (2-3) | 7 |
| 16 | CS | 12 | System Programming | 3(2-3) | 6 |
| 17 | CS | 5 | Numerical and Symbolic Computation **(18/130)** | 3(3-0) | 7 |
| **Elective Computer Science Courses** (Not Limited to the List below) | | | | | |
| | | **Area** | **Course** | | |
| 18 | CS | **Numerical Computation** | Numerical Computing | 3 (2-3) | 5 |
| 19 | CS | **Computer Graphics** | Computer Graphics | 3 (2-3) | 6 |
| 20 | CS | **Software Engineering** | Software Engineering-II | 3 (3-0) | 5 |
| 21 | CS | **Languages and Translators** | Compiler Construction | 3 (2-3) | 7 |
| 22 | CS | | Principles of Programming Languages | 3 (2-3) | 4 |
| 23 | CS | **Computer/ Communication Networks** | Data Communication | 3 (3-0) | 6 |
| 24 | CS | | Distributed Computing | 3 (2-3) | 6 |
| 25 | CS | | Data and Network Security | 3(3-0) | 7 |
| 26 | CS | | Wireless Networks | 3(2-3) | |
| 27 | CS | **Visual Programming** | Visual Programming | 3 (2-3) | 7 |

| | | | | | |
|---|---|---|---|---|---|
| 28, 29 | CS | **Computer Architecture** | Computer Architecture<br><br>Microprocessor Interfacing | 3(2-3)<br><br>3(2-3) | |
| 30 | CS | **Signal Processing** | Digital Signal Processing | 3(2-3) | |
| 31 | CS | | Digital Image Processing | 3(2-3) | |
| 32 | CS | **Web Engineering** | Web Engineering | 3 (2-3) | 5 |
| 33 | CS | **Systems Software** | System Programming | 3 (2-3) | 7 |
| 34 | CS | **Database Systems** | Distributed Database Systems | 3 (2-3) | 7 |
| 35 | CS | | Data Warehousing | 3(2-3) | 6-7 |
| 36 | CS | **Human Computer Interaction (HCI)** | Human Computer Interaction | 3(2-3) | 6-7 |
| | | | **(21/130)** | | |

**Computing Requirements-Supporting Sciences 12 Credit Hours (Refer to Computing part)**

| Required Supporting Courses | | | | | |
|---|---|---|---|---|---|
| 5 | PH | 38 | Physics – II | 3 (3-0) | 4 |
| 6 | ST | | Multivariate Calculus | 3(3-0) | 4 |
| 7 | ST | - | Differential Equations **(9/130)** | 3 (3-0) | 5 |

**Computing Requirements-General Education 12 Credit Hours (Refer to Computing part)**

---

| **University Electives-Recommended Courses (**Not Limited to the List below) | | | | | |
|---|---|---|---|---|---|
| 46 | MG | - | Financial Accounting | | |
| 47 | MG | - | Financial Management | 3 (3-0) | 4 |
| 48 | MG | - | Human Resource Management | 3 (3-0) | 5 |
| 49 | MG | - | Marketing | 3 (3-0) | 6 |
| 50 | SS | - | Economics | 3 (3-0) | 7 |
| 51 | PS | - | Psychology | 3 (3-0) | 6 |
| 52 | SS | - | International Relations | 3 (3-0) | 7 |
| 53 | SS | - | Foreign Language (French, German, etc.) | 3 (3-0) | 7-8 |
| 54 | SS | - | Philosophy | 3(3-0) | 6-8 |
| | | | **(18/130)** | | |

***Courses Profiles and contents of BS (CS) Programmes***

## BS Computer Science Courses (Core)

| **Course Name:** Computer Organization and Assembly Language | |
|---|---|
| **Course Structure:** Lectures: 2 / Labs: 3 | **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:** Objectives and Perspectives of Assembly Language, Microprocessor Bus Structure: Address, Data and Control, Memory Organization and Structure (Segmented and Linear Models), Addressing Modes, Introduction to the Assembler and Debugger, 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. |
| **Reference Material:**<br>*Assembly Language for Intel-Based Computer,* Third Edition, 1999, by Kip R. Irvine. Prentice-Hall Publishing, 1999 |

| | |
|---|---|
| **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. | |
| **Reference Material:**<br>*Introduction to Computer Theory*, Denial Cohen, John Wiley & Sons, Inc.<br>*Introduction to Automata Theory, Languages and Computation*, J Hopcraft, D. Ullman.<br>*Languages and Machines, An Into to the Theory of Comp. Sc.*, 2/e Thomas A. Sudkamp, Addison Wesley. | |

| | |
|---|---|
| **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. | |
| **Reference Material:**<br>*Introduction to Algorithms,* T. H. Cormen, C. E. Leiserson, and R. L. Rivest, MIT | |

| |
|---|
| Press, McGraw-Hill, New York, NY, 1990. |

| | |
|---|---|
| **Course Name:** Artificial Intelligence | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Discrete 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:**<br>*Artificial Intelligence* by Luger, 4th edition, Pearson Education. | |

| | |
|---|---|
| **Course Name:** Numerical Computing | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Visual Programming | |
| **Objectives:** On completion of this unit, students will be able to demonstrate programming proficiency using structured programming techniques in suitable programming languages and implement numerical solutions using computer-based techniques. | |
| **Course Outline:** Mathematical Preliminaries, Solution of Equations in one variable, Interpolation and Polynomial Approximation, Numerical Differentiation and Integration, Initial Value Problems for Ordinary Differential Equations, Direct Methods for Solving Linear Systems, Iterative Techniques in Matrix Algebra, Solution of non-linear equations. Approximation Theory. Eigenvalues and Eigenvector computation. | |
| **Reference Material:**<br>*Elements of Numerical Analysis*, Dr. Faiz, M. Afzal | |

| | |
|---|---|
| **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 O 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 Binaryble format (ELF), Dynamic shared objects, Unix Kernel Programming (Ring O), Unix Device Architecture (Character & Block Devices), Device Driver Development, Enhancing Unix Kernel. | |
| **Reference Material:**<br>*The UNIX Programming Environment,* B. Kernighan & R. Pike Prentice-Hall, 1984.<br>*System Software,* Leland L. Beck*,* Addison-Wesley Longmsan, 1990, ISBN: 0-201-50945-8. | |

## *BS Computer Science Courses (Elective)*

| | |
|---|---|
| **Course Name:** Compiler Construction | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Theory of Automata | |
| **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:**<br>Compiler Design and Construction*, by Alfred V. Aho, Ravi Sethi, Hardcover 2nd edition,*<br>*1987, Van Nostrand Reinhold; ISBN: 0317636367.* | |

| | |
|---|---|
| **Course Name:** Software Engineering-II | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Data Structures, Software Engineering–I | |
| **Objectives:** The students will study techniques for software verification, validation and testing. They would also study reliability and performance issues in software design and development. | |
| **Course Outline:** Software verification and validation: Techniques are introduced to evaluate software correctness, efficiency, performance and reliability, integration of these techniques into a verification and validation plan. Technical reviews, software testing, programme verification, prototyping, and requirement tracing. Attitude of industry toward reliability and performance. | |
| **Reference Material:**<br>*Software Engineering: A Practioner's Approach,* Roger Pressman, McGraw-Hill, 2001.<br>*Software Engineering,* Ian Sommerville, Addison-Wesley 2001, ISBN: 0-201-39815-X. | |

| | |
|---|---|
| **Course Name:** Software Project Management | |
| **Course Structure:** Lectures: 3 / Lab 0 | **Credit Hours:** 3 |
| **Prerequisites:** Data Structures, Software Engineering–I | |
| **Objectives:** To develop ability to plan and manage software development projects successfully, maximizing the return from each stage of the software development life cycle. | |
| **Reference Material:**<br>*Software Project Management,* Richard H. Thayer, Wiley IEEE Press 2002, ISBN 0-7695-1199-6.<br>*Software Engineering:* A Practitioner's Approach, 4th edition, Roger S. Pressman, McGraw-Hill Higher Education, ISBN: 0070521824. | |

| | |
|---|---|
| **Course Name:** Telecommunications 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:**<br>*Introduction to telecommunications Network Engineering,* 2nd edition, T. Aattalainen, Artech House 2003, ISBN: 1580535003. | |

| |
|---|
| *Fundamentals of Telecommunication Networks,* T. Saadawi, Wiley US, ISBN: 0471515825.<br>*Telecommunication Systems,* P. G. Fonteolliet, Artech House 1991. |

| | |
|---|---|
| **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, 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. | |
| **Reference Material:**<br>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.<br>*2. Computer Graphics,* F.S.Hill, Maxwell MacMillan ISBN: 0-02-354860-6. | |

| | |
|---|---|
| **Course Name:** Network Management | |
| **Course Structure:** Lectures: Labs: | **Credit Hours: 2/3** |
| **Prerequisites:** Computer Networks | |
| **Objectives:** The objective of this course is to that students learn and a have hands-on experience of establishing, managing, troubleshooting and maintaining computer networks. | |
| **Course Outline:** Introduction, Overview of Network operating system, Setting up a network server, Setting up network clients, Network design issues, Network client administration, Workgroup and Domain concepts. System Administration Basics, Network Services, Monitoring and logging of various operating system events, security applications and general system events, Monitoring wide variety of system objects, Diagnosing and Troubleshooting hardware, networking and various operating system services, System configuration including screen display, network services, bindings, protocols, servers, services and system devices, User and group management and services used to manage user access to resources, Remote administration, Management of key processes, Network services administration including e-mail, internet, web and ftp, Heterogeneous network environment, Firewall administration, Controlling access to the machines. | |
| **Reference Material:**<br>*Network Management — A Practical Perspective,* 2nd Edition, Allan Leinwand and Karen Fang Conroy, Addison Wesley Professional, ISBN: 0-201-60999-1. | |

| **Course Name:** Distributed Database Systems | |
|---|---|
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Data Base Systems | |
| **Objectives:** The student will learn the design, algorithms and techniques involved in distributed database system and their implementation. | |
| **Course Outline:** Advanced data models. Conceptual Database design. Concurrency control techniques. Recovery techniques. Query processing and optimization. Integrity and security. Client-Server architecture. Distributed database systems. Current trends in database systems. Database machines. | |
| **Reference Material:**<br>Distributed Databases: Principles and System, *Ceri and Pelagatti McGraw-Hill Book Company 1984, ISBN: 0-07-010829-3*. | |

| **Course Name:** Visual Programming | |
|---|---|
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Data Structures, Data and Network Security | |
| **Objectives:** To development applications using various tools and APIs in visual programming. | |
| **Course Outline:** Introduction to Windows programming, Use of Windows API, MFC Class hierarchy, Class Wizard, Application Wizard and Application Studio, Graphics Device Interface, Menus, document view architecture, Multiple Views, files and archiving mechanisms, converting Windows programmes to MFC, Sub-classing controls. | |
| **Reference Material:**<br>*MFC from the Ground Up.*<br>*Windows 98 API Programming.*<br>*VC++ A complete References.* | |

| **Course Name:** Multimedia Technologies | |
|---|---|
| **Course Structure:** Lectures: Labs: | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** This course is aimed at exposing students to the current and future trends in Multimedia design and development. There is a huge amount of activity going on in this field with a big market all over the world, and new tools and technologies emerge quickly. Students shall learn them and familiarize themselves with the solution development using these tools. | |
| **Course Outline:** Introduction to Multimedia Programming, Scope of Multimedia Programming, convention and trends, Media types used in current applications (including digital video, audio, and graphics). System level issues of performance synchronization, storage and server schemes, dynamic interactivity, hyper linking, multimedia device control, distributed media development and delivery, non-standard media and programming frame works. Introduction to Multi-media Networks. | |
| **Reference Material:**<br>*Multimedia Systems Design,*1/e, Andleigh, P.K. and Thakrar, K., Prentice Hall. | |

| | |
|---|---|
| **Course Name:** Web Engineering | |
| **Course Structure:** Lectures: Labs: (2-3) | **Credit Hours:** 3 |
| **Prerequisites:** Data Warehousing | |
| **Objectives:** Design and implementation of web based applications. | |
| **Course Outline:** Overview of Protocols: TCP/IP, HTTP, Overview of 3-tier Architecture, Web Based Applications Architecture. Developing Front End Applications: Front End Development Tools, HTML, DHTML, Scripting (Java Script, Jscript, Vbscript), Java Applets, ActiveX. | |
| **Reference Material:**<br>*Web enabled Commercial Application Development Using…HTML, DHTML, JavaScript, Perl, CGI,* Ivan Bayross, BPB Publications. | |

| | |
|---|---|
| **Course Name:** Modern Programming Languages | |
| **Course Structure:** Lectures: Labs: | **Credit Hours: ¾** |
| **Objectives:** As a senior level course, this course aims at uplifting students' approach and thinking of software development process and tools in general and programming in particular**.** | |
| **Course Outline:** Developing Back End Applications: Java Servlets, CGI/Perl Programming, Cold Fusion, Gateway Interface. Database Connectivity: DBC, ODBC. Performance & Security Issues: E-Commerce Application Security, Presentations, Projects. | |
| **Reference Material:**<br>Modern Programming Languages: A Practical Introduction, *Adam Webber, ISBN: 1-887902-76-7.* | |

| | |
|---|---|
| **Course Name:** Advanced Computer Networks | |
| **Course Structure:** Lectures: Labs: | **Credit Hours:** 3 |
| **Prerequisites:** Computer Networks | |
| **Objectives:** A comprehensive understanding and analysis of the issues of computer networks and the known issues and trade-off scenarios. Future trends and modern protocols and standards are also covered in this course. | |
| **Course Outline:** Multi-access Communications, Introduction to Layered Network Architecture, Inter Networking, Advanced Topics in flow Control, Congestion Control and rooting, Protocol Performances in Lan and Wan Environment, Network Privacy and Security. | |
| **Reference Material:**<br>*1. Computer networks: a systems approach*, Larry Peterson, Bruce Davie, Princeton Univ., Princeton.<br>2. *Computer Networking: A Top-Down Approach Featuring the Internet,* 2/e, James F Kurose, Keith W Ross, Addison Wesley 2003. ISBN: 0-201-97699-4 | |

| | |
|---|---|
| **Course Name:** Advanced Software Engineering | |
| **Course Structure:** Lectures: Labs: | **Credit Hours:** 3 |
| **Prerequisites:** Software Engineering-I | |
| **Objectives:** This consummates the knowledge and skills learnt in first Software Engineering course. The complete software engineering cycle is covered with current methodologies and techniques. Student shall also learn the key industrial standards and practices in software engineering today. | |
| **Course Outline:** System Development using Formal Techniques, Algebraic specification, Abstract model specification, Verification: Proof Systems, Proof Techniques, Proof obligations, Design: Data refinement, operation refinement, Design decomposition. Software Reliability and Metrics. Macro models: productivity, effort. Defect models: Software reliability, Failures and fault, Software reliability modelling. Simple model, Markov modelling, Parameter estimation, Comparison of models. | |
| **Reference Material:**<br>*Developing Software with UML,* Bernd Oesterich, Addison-Wesley.<br>*Developing Object Oriented Software,* OOTC, Prentice Hall.<br>*Unified Modeling Language Reference Manual, The (2nd Edition),* James Rumbaugh, Ivar Jacobson, Grady Booch, Pearson Higher Education 2004. ISBN: 0321245628. | |

## *BS Computer Science Courses (Required Supporting Courses)*

| | |
|---|---|
| **Course Name:** Multivariable Calculus | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Calculus and Analytical Geometry | |
| **Objectives:** Cover Secondary level to advanced topics in Calculus | |
| **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:**<br>*Calculus and Analytical Geometry By Swokowski, Olinick and Pence.* | |

| | |
|---|---|
| **Course Name:** Differential Equations | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Multivariate Calculus | |
| **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:**<br>*Advanced Engineering Mathematics,* Michael Greenberg, ISBN: 0133214311.<br>*Advanced Engineering Mathematics,7/e,* Erwin Kreyszig. John Wiley & Sons 1992 ISBN: 0471553808.<br>*A First Course in Differential Equations,* Zill, Prindle, Weber and Schmidt. Brooks/Cole Publishing, 1996. ISBN: 0534955746.<br>*Differential Equations With Boundary-Value Problems*, Dennis G. Zill, Michael R. Cullen. Brooks/Cole Publishing, 1996. ISBN: 0534955800.<br>*Elementary Differential Equations With Applications*, C. H. Edwards, David E. Penney. Prentice Hall, 1993. ISBN: 0132534282. |

| | |
|---|---|
| **Course Name:** Physics-II | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** |
| **Prerequisites:** None | |
| **Objectives:** Cover the fundamental topics in mechanics. | |
| **Course Outline:** Vector Motion: position, velocity, and acceleration vectors. Newton's Laws. Projectile Motion, Uniform circular motion, and application. Rotational motion: constant angular acceleration, torque. Momentum: linear and angular momentum, two-body collisions, conservation of momentum. System of Particles: motion of complex objects, centre of mass of solid objects. Work and Energy: power, kinetic and potential energy, conservative systems. | |
| **Reference Material:**<br>*University Physics, 8th edition*, Hugh Young, Addison Wesley.<br>*Physics, 2nd ed.* (199'')), McGraw Hill, ISBN: 0-07-023461-2.<br>*Physics,* Resnick, Halliday, Krane, Wiley Sons. | |

## *BS Computer Science Courses (Elective General Courses)*

| | |
|---|---|
| **Course Name:** Financial Accounting | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Introduction to Accounting and its concepts. Recording Business Transactions: Journal, Ledger, Trial Balance. Preparation of Financial Statements: Balance Sheet, Income Statement, Completion of Accounting Cycle: Adjustments, Closing, Work Sheet Accounting for purchase and sales of | |

| |
|---|
| merchandise. Receivable and payable, Inventories, Payroll Systems. Plant and Equipment: Acquisition, Depreciation, Disposal. Corporations: Organization and stock-holders equity, Operations, Earning per share and dividends. |
| **Reference Material:**<br>*Accounting: The Basis for Business Decisions* by Meigs & Meigs, 10th Edition. |

| | |
|---|---|
| **Course Name:** Financial Management | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Introduction to Financial Management, Concepts and Models in Valuation, The time value of money, Fundamentals of risk and portfolio analysis, Valuation of stocks and bonds, The capital Asset Pricing Model, the Arbitrage Pricing Model and other valuation models. The Cost of Capital: Capital structure and Dividend Policy, The cost of capital, Capital structure theory, Capital structure policy and optimal capital structure, Internal financing and dividends policy Capital Budgeting: The basis of capital budgeting, The determination and use of cash flow, Mutually exclusive investments and capital rationing, Annual equivalent cost and replacement decisions, Risk analysis and the optimal capital budget, Islamic guidelines for financial management: The rational of prohibition of interest, Alternate capital structure, Capital Budgeting in an Interest free economy, working Capital Management in 100% equity capital structure. | |
| **Reference Material:** *Financial Management* by Charles H. Gibson. | |

| | |
|---|---|
| **Course Name:** Human Resource Management | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** An overview of Human Resource Management and Human Resource Manager. The Environment of Human Resource Management, external and Internal Environment. Equal Employment Opportunity and Affirmative Action. Job Analysis: A Basic Human Resource Tool. Human Resource Planning, Recruitment, and Selection. Organization Change and Human Resource Development. Corporate Culture and Organization Development. Career Planning Development. Performance Appraisal. | |
| **Reference Material:**<br>*Managing Human Resource* by Wayne F. Cascio. | |

| | |
|---|---|
| **Course Name:** Organizational Behaviour | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Fundamentals of Organizational Behaviour, Behavioural Science and Organizational Behaviour, Individual Behaviour in Organizations, Personality, perception and attitudes, Learning and reinforcement, Motivation, Group Behaviour in Organizations, Group dimensions in organizations, Group dynamics, Leadership, Organizational Structure and Organizational Behaviour, Organizational design, Job design, Stress and work, Organizational Processes, The decision-making process, The communication process, Performance | |

| |
|---|
| appraisal process, Special Issues in Organizational Behaviour, Management of conflict and change, Organizational development, Impact of computer technology. |
| **Reference Material:**<br>*Organizational Behaviour* by Fred Luthans. |

# Curriculum for MS (Computer Science)

The participants were of the opinion that MS (Computer Sciences) programme should be designed in such a manner that it is the second degree in the field of Computer Sciences. The following main areas were discussed:

- **Degree Equivalency**
- **Eligibility Criteria**
- **Structure of MS (Computer Sciences)**
- **Breadth and Depth Integrated Based Programme**
- **Theoretical Computer Sciences Based Core Structure**
- **Research Based Programmes**

Some changes were proposed in the last meeting's recommendations for MS programme. The complete detail regarding proposed MS (Computer Sciences) Programme is available in the following pages which is based on the following structure:

- 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 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:

| **Category or Area** | **Credit Hours** |
|---|---|
| Core | 12 |
| Minimum Electives in Area of Specialization | 6 |
| Electives | 6 |
| Thesis | 6 |
| **Total Credit Hours** | **30** |

**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.

The following are relevant details:

- 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 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:

| **Category or Area** | Credit Hours |
|---|---|
| Core | 12 |
| Minimum Electives in Area of Specialization | 6 |
| Electives | 6 |
| Thesis | 6 |
| **Total** | **30** |

***Eligibility***

- BS (CS) 4 Years Degree Programme (min 120 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 120 credit hours.
- 16 year Science and Engineering graduates are eligible but they have to cover deficiency

## Courses Requirements:

*Core courses*

| # | Code | Course Title | Credit hours | Semester |
|---|---|---|---|---|
| 1 | CS | Theory of Computation | 3 | 1 |
| 2 | CS | Advanced Algorithm Analysis | 3 | 1 |
| 3 | CS | Advanced Operating Systems | 3 | 1 |
| 4 | CS | Advanced 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 Engineering | 3 | | CS | Design of Intelligent Systems | 3 |
| CS | Topics in Software Engineering | 3 | | CS | Machine Learning | 3 |
| CS | Object Oriented Software Engg. | 3 | | CS | Neural Networks | 3 |
| CS | Software Quality Assurance | 3 | | CS | Mathematical Reasoning | 3 |
| CS | Requirements Engineering | 3 | | CS | Decision Support Systems | 3 |
| CS | Distributed Computing | 3 | | CS | Computer Vision | 3 |
| | | | | | | |
| | **Databases Systems** | | | | **Computer Architecture** | |
| CS | Advanced DBMS | 3 | | CS | Advanced Logic Design | 3 |
| CS | Data Warehousing | 3 | | CS | Digital System Design | 3 |
| CS | Object Oriented Databases | 3 | | CS | Integrated Circuit | 3 |
| CS | Web-Based DBMS | 3 | | CS | Design Verification | 3 |
| CS | Topics in DBMS | 3 | | | | |
| | | | | | | |
| | **System Engineering** | | | | **Multimedia & Graphics** | |
| CS | Advanced Computer Architecture | 3 | | CS | Advanced Computer Graphics | 3 |
| CS | Digital Signal Processing | 3 | | CS | Multimedia & Hypermedia System | 3 |
| CS | Switching and Fault Diagnosis | 3 | | CS | Virtual Reality | 3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| CS | Parallel & Distributed Computing | 3 | | CS | Human Computer Interface | 3 |
| CS | Control Systems and Robotics | 3 | | CS | Geographical Information Systems | 3 |
| CS | Real Time Operating Systems | 3 | | CS | Computer Animation | 3 |
| | | | | | | 3 |
| | **Computer Networks** | | | | **Computer Science Education** | |
| CS | Advanced Networking | 3 | | CS | Educational Technology | 3 |
| CS | Network Security | 3 | | CS | Multimedia and Hypermedia Sys | 3 |
| CS | Topics in Computer Networking | 3 | | CS | Computer Aided Instructions | 3 |
| CS | Network Transport Protocols | 3 | | CS | Web Based Education Systems | 3 |
| CS | Network Administration | 3 | | CS | Measurement of Learning | 3 |
| CS | Wireless Networks | 3 | | CS | Topics in Comp Science Education | 3 |
| CS | Network Performance Evaluation | 3 | | | | |
| | | | | | | |
| | **Compiler Design & Construction** | | | | **Human Computer Interaction** | |
| CS | Theory of Programming Languages | 3 | | CS | Intelligent User Interfaces | 3 |
| CS | Advanced Compiler Design–I | 3 | | CS | Multimedia Database | 3 |
| CS | Advanced Compiler Design–II | 3 | | CS | Information Retrieval Techniques | 3 |
| | | | | CS | Rich Internet Applications | 3 |

## *Model Programme: Semester-wise Plan*

### *MS (CS)*

Semester 1 (12 credit hrs)

| | | **Subjects** | **Credit Hrs** |
|---|---|---|---|
| 1 | CS | 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) | 3 |
| | | | Total: 3 |

**Total (all semesters)=30**

## Course Description and Profiles:

**Core courses:**

| **Course Name:** Theory of Computation | |
|---|---|
| **Course Structure:** Lectures: 3 Labs: 0 | **Credit Hours:** 3 |
| **Course Description:** 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:**<br>Michael Sipser, *Introduction to the Theory of Computation*, First Edition, 1997, PWS Publishing Company.<br>Christos Papadimitriou, *Computational Complexity*, 1994, Addison-Wesley.<br>John Hopcroft and Jeffrey Ullman, *Introduction to Automata Theory, Languages, and Computation*, 1979, Addison-Wesley. (or the second edition).<br>Tao Jiang, Ming Li, and Bala Ravikumar, Formal models and Computability, in *Handbook of Computer Science*, CRC Press, 1996.<br>T.H. Cormen, et al., *Introduction to Algorithms,* MIT Press and McGraw-Hill Book Co., 1990.<br>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 |
| **Course Description:** 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:**<br>1. *Introduction to Algorithms,* by T. Cormen, C. Leiserson, and R. Rivest. MIT Press and McGraw-Hill.<br>2. *Computers and Intractability, Guide to the Theory of NP-Completeness*, by M. Garey and D. Johnson.<br>*Computation*, 1979, Addison-Wesley. (or the second edition).<br>Tao Jiang, Ming Li, and Bala Ravikumar, Formal models and Computability, in *Handbook of Computer Science*, CRC Press, 1996. | |

T.H. Cormen, et al., *Introduction to Algorithms,* MIT Press and McGraw-Hill Book Company, 1990.
Peter Linz*, An Introduction to Formal Languages and Automata*, ISBN: 0-669-17342-8.

| **Course Name:** Advanced Operating Systems | |
|---|---|
| **Course Structure:** Lectures: 3 Labs: 0 | **Credit Hours:** 3 |
| The class covers advanced topics in computer operating systems with a special emphasis on distributed computing, and the services provided by distributed operating systems. Important topics include naming, security, remote procedure call, networks, concurrency, transactions, parallel computing, shared memory, message passing, and scale. | |
| **Text Books/Reference Books:**<br>*Distributed Systems: Concepts and Design* by Coulouris, Dollimore, and Kindberg, 3rd Edition. | |

| **Course Name: Advanced Computer Architecture** | |
|---|---|
| **Course Structure:** Lectures: 3 Labs: 0 | **Credit Hours:** 3 |
| **Course Description:** Design and evaluation of modern uniprocessor computing systems. Evaluation methodology/metrics and caveats, instruction set design, advanced pipelining, instruction level parallelism, prediction-based techniques, alternative architectures (VLIW, Vector and SIMD), memory hierarchy design and I/O. Case studies. | |
| **Text Books/Reference Books:**<br>John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd Edition, Morgan Kaufmann Publishers, 2002.<br>Andrew S. Tanenbaum, *Modern Operating Systems*, 2nd Edition, Prentice Hall, 2001.<br>John Hennessy and David Patterson, *Computer Organization and Design: The Hardware/Software Interface*, 2nd Edition, Morgan Kaufman Publishers. | |

## Selected Electives courses:

| **Course Name:** Digital Signal Processing | |
|---|---|
| **Course Structure:** Lectures: 3 Labs: 0 | **Credit Hours:** 3 |
| **Course Description:** 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:**<br>*Discrete-Time Signal Processing,* 2nd edition Alan V. Oppenheim and Ronald W. | |

| |
|---|
| Schafer, Prentice-Hall. |

| | |
|---|---|
| **Course Name:** Parallel and Distributed Computing: | |
| **Course Structure:** Lectures: 3 Labs: 0 | **Credit Hours:** 3 |
| **Course Description:** 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:**<br>B. Wilkinson and M. Allen, *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, 1/e,* Prentice Hall, 1999.<br>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 Description:** 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:**<br>R.C. Dorf, Modern Control Systems, 7th (1995), 8th (1998) or 9th (2001) Edition, Addison-Wesley.<br>C.C. Bissell, Control Engineering, 2nd Edition, 1994, Publisher: Chapman & Hall.<br>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 Description:** 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:**<br>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 Description:** 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.<br>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:**<br>James F. Kurose and Keith W. Ross, "Computer Networking – A Top-Down Approach Featuring the Internet", Addison Wesley.<br>Coulouris, Dollimore, Kindberg, "Distributed Systems – Concepts and Design", Addison Wesley.<br>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 Description:**<br>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:**<br>W. Stallings, *Cryptography and Network Security*, Prentice Hall PTR, Upper Saddle | |

River, NJ, 2003.
C. Kaufman, R. Perlman, M. Speciner, *Network Security: Private Communication in a Public World* – Prentice Hall PTR, Upper Saddle River, NJ, 2002.
M. Bishop, *Computer Security: Art and Science* – Addison-Wesley, 2003.
D. Stinson, *Cryptography: Theory and Practice,* CRC Press, Boca Raton, FL, 1995.
Richard A. Mollin, *An Introduction to Cryptography*, Chapman and Hall/CRC, 2001.
B. Schneier, *Applied Cryptography,* John Wiley and Sons, NY, 1996.
A. Menezes, P. Oorshcot, 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 Description:** 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:<br>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:**<br>Puzmanov, *Switching and Routing*, Addison Wesley, 2002.<br>Garica and Widjaja, *Communication Networks: Fundamentals Concepts and Key Architectures,* McGraw-Hill, 2001.<br>Peterson and Davie, *Computer Networking a Systems Approach*, 3rd Edition, Morgan Kaufman, 2003.<br>William Stallings, *High-Speed Networks: TCP/IP and ATM Design Principles,* Prentice Hall; 1998, ISBN: 0135259657.<br>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 Description:** 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:**

*Information Technology Project Management.* (2002) Course Technology. ISBN: 0-619-03528-5.

- *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 Description:** 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:**<br>W. Stallings, "Wireless Communications and Networks", Prentice Hall, 2002.<br>T.S. Rappaport, "Wireless Communications: Principles & Practice", Second Edition, Prentice Hall, 2002.<br>J. Schiller, "Mobile Communications", Addison Wesley, 2000.<br>V.K. Garg, "IS-95 CDMA and cdma 2000", Prentice Hall PTR, 2000.<br>J.P. Castro, "The UMTS Network and Radio Access Technology - Air Interface Techniques for Future Mobile Systems", Wiley, 2001.<br>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 Description:** 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:**<br>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 Description:** 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:** | |

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

| **Course Name:** Advanced Compiler Design I: | |
|---|---|
| **Course Structure:** Lectures: 3 Labs: 0 | **Credit Hours:** 3 |
| **Course Description:** 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:**<br>*Compilers: Principles, Techniques, and Tools*, Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Addison-Wesley, 1988.<br>*Advanced Compiler Design & Implementation*, Steven S. Muchnick, Morgan Kaufmann, 1997.<br>*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 Description:** 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:**<br>*Compilers: Principles, Techniques, and Tools*, Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Addison-Wesley, 1988.<br>*Advanced Compiler Design & Implementation*, Steven S. Muchnick, Morgan Kaufmann, 1997.<br>*Building an Optimizing Compiler*, Robert Morgan, Butterworth-Heinemann, 1998. | |

| **Course Name:** Intelligent User Interfaces: | |
|---|---|
| **Course Structure:** Lectures: 3 Labs: 0 | **Credit Hours:** 3 |
| **Course Description:** The increasing complexity of software and the proliferation of information makes intelligent user interfaces increasingly important. The promise of interfaces that are knowledgeable, senstitive 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:** *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 Description:** 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:**<br>Principles of Multimedia Database Systems, by V.S. Subrahmanian, Morgan Kaufmann Publishing Company, San Fransisco, CA. 1998. ISBN: 1558604669.<br>Principles of Database Query Processing for Advanced Applications (Morgan Kaufmann Series in Data Management Systems), by Clement T. Yu, Weiyi Meng, 1998. ISBN: 1558604340.<br>Databases and Transaction Processing, An Application-Oriented Approach, Philip M. Lewis, Arthur Bernstein, and Micheal Kifer. Addison Wesley Publishers, 2002. ISBN: 0201708728. | |

| **Course Name:** Rich Internet Applications: | |
|---|---|
| **Course Structure:** Lectures: 3 Labs: 0 | **Credit Hours:** 3 |
| **Course Description:** 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. | |

# Software Engineering Curricula 2004

## National Curriculum Revision Committee — Software Engineering

The National Curriculum Revision Committee for Software Engineering (NCRC-SE) met on 1-3 January, 2004 to develop the vision for Software Engineering education and curriculum for software engineering programmes. The Committee met again on 12-14 April, 2004 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. Aftab A. Maroof — **Convener**
   Professor and Director
   National University of Computer & Emerging Sciences
   FAST House, Rohtas Road, G-9/4
   Islamabad

2. Dr. Imdad Ali Ismaili — **Secretary**
   Professor
   Institute of Information Technology
   University of Sindh
   Jamshoro

3. Dr. Nazir Ahmed Sangi — **Member**
   Professor/Dean
   Faculty of Science
   Allama Iqbal Open University
   Islamabad

4. Dr. Zubair Ahmed Shaikh — **Member**
   Professor
   National University of Computer & Emerging Sciences
   Shah Latif Town
   Karachi-75030

5. Dr. Fakhar Lodhi — **Member**
   Professor
   National University of Computer and Emerging Sciences
   Faisal Town
   Lahore

6. Dr. Jafar Rehman — **Member**
   Dean
   Faculty of Engineering & Computer Science
   M.A Jinnah University
   Islamabad-Karachi

7. Dr. Naveed Ikram **Member**
   Dean
   Faculty of Information Technology
   Riphah International University
   Islamabad

8. Dr. M. Ashraf Chughtai **Member**
   Professor, Department of Electrical Engineering
   University of Engineering & Technology
   Lahore

9. Dr. Aqil Burny **Member**
   Chairman
   Department of Computer Science
   University of Karachi
   Karachi

10. Dr. Abdul Qadir **Member**
    Professor
    M. A. Jinnah University
    Islamabad

11. Dr. Abdul Wahab Ansari **Member**
    Professor
    Institute of Information Technology
    University of Sindh
    Jamshoro

12. Dr. Javeed Qamar **Member**
    Dean
    Dada Bhai Institute of Technology
    Karachi

13. Dr. Sabir Hussain Usmani **Member**
    Chairman, Department of Electronic Engineering
    Balochistan University of Information Technology
    & Management Sciences
    Jinnah Town, Sumung Road
    Quetta

14. Mr. Muhammad Sibghatullah Siddiqui **Member**
    Chairman
    Department of Computer Science
    Muhammad Ali Jinnah University
    22-E, Block-6, PECHS
    Karachi

15. Mr. Tahseen Hafiz **Member**
    Associate Professor
    Mehran University of Engg. & Technology
    Jamshoro

16. Prof. Hamid Jan **Member**
    Director, Institute of Management Sciences
    Sarhad University, 2, Sir Syed Road
    Peshawar Cantt.

17. Syed Zubair Ahmed **Member**
    Senior Scientific Officer
    Computer Division, PINSTECH, NILORE
    Islamabad

18. Mr. Sajjad Hussain **Member**
    Assistant Professor
    Faculty of Management Sciences
    International Islamic University
    Islamabad

19. Mr. Hamid Raza Malik **Member**
    Assistant Professor
    Institute of Management Technology
    14-Mabar Block, New Garden Town
    Lahore

20. Mr. Muhammad Naeem **Member**
    Assistant Professor
    Department of Computer Science
    University of Peshawar
    Peshawar

21. Mrs. Nosheen Fayyaz **Member**
    Lecturer,
    Institute of Business Management & Computer Sciences
    NWFP Agricultural University
    Peshawar

22. Mr. Irfan Ghani **Member**
    Assistant Professor
    Balochistan University of Information Technology
    & Management Sciences
    Jinnah Town, Sumung Road
    Quetta

23. Mr. Nisar Ahmed **Member**
Senior Lecturer, Baqai Medical University
51-Deh Tor, Near Toll Plaza, Super Highway
Karachi

24. Syed Asif Ali **Member**
Lecturer, Bahria Institute of Management
and Computer Sciences
Karachi

25. Mr. Shahbaz Rasheed Bhatti **Member**
ORIX Leasing, OICC Building
Karachi

## The Discipline of Software Engineering

Software Engineering is the discipline of creating high-quality software in a systematic, controlled and efficient manner, and maintaining it affordably. It involves the application of engineering concepts, techniques, and methods to the development of software 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 Physics[1]. 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 Parnas, is to *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. 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."

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

**Vision**

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

> *The SE education in Pakistan will focus on imparting to students the knowledge and training which should enable them to harmonize theory with practice, concept with application, and problem with solution. It will prepare them to ably apply engineering principles, processes and practices to software components and systems, and their maintenance. The programme will also, in addition to students' professional growth, attend to development of their personal and interpersonal skills. It will help students to enhance their ability in oral and written communication, and their adaptability to group-work environments. The programme will inculcate among students a strong sense of civic, professional and ethical responsibility. The programme will strive to develop in the professionals 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. 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 sprit of engineering criteria above. It is noted that recent 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)[2], Software Engineering Education Knowledge (SEEK)[3], Computing Curriculum—Software Engineering[4] (CCSE) and Computing Curriculum 2004—Overview Report[5].

---

[2] *Guide to Software Engineering Body of Knowledge,* IEEE Trial Version 1.0, May 2001.

[3] *Computing Curricula—Software Engineering Volume, Software Engineering Education Knowledge (SEEK)*, Final Draft, April 30, 2003.

[4] *Computing Curriculum—Software Engineering,* (CCSE) Public draft 3.1, February 6, 2004 by ACM/IEEE

[5] Computing Curriculum 2004—Overview Report, A guide to undergraduate degree programmes in computing

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 forth years. Lab component should inculcate among students an industrious approach and practice of problem solving. Good SE practices must be nurtured all along 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 of reasonably broad exposure of 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 gained in a wide variety of courses to solve realistic problems in a team-based environment.

## Software Engineering Degree Programmes

**Nomenclature**

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

*Bachelor of Science in Software Engineering — BS Software Engineering — BS (SE)*
*Master of Science in Software Engineering — MS Software Engineering — MS (SE)*
*Doctor of Philosophy in Software Engineering — Ph.D. Software Engineering — Ph.D.*

**Duration of Programmes**

The committee defined a credit hour as 15 lecturing hours in a course offered in a particular semester. It was agreed that 2 to 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 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 physics and 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.

# 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, methodology and tools in the modern software development environments.

Graduates of such programmes 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 software development. They will have knowledge of and experience with software product engineering and engineering management and an understating 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 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 principles of software engineering practice and process, subject to realistic constraints.
- be able to model, analyze, document and track system requirements.
- be able to design, implement and maintain software systems.
- be able to verify and validate the software systems.
- have the awareness of current industry standards and practices.
- be able to work in one or more application domains.
- understand and apply principles of team process.
- be able to understand and apply software project management skills: estimation, costing, planning, deployment and tracking of resources.
- have strong communication 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 systematically achieve the objectives set above for the programme. It has been structured to suit the needs of the students, the

[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 .

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 into 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 sure way. Students will also be imparted the skills and techniques which derive essence from basic sciences like mathematics and physics. These areas will be taken care of 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.

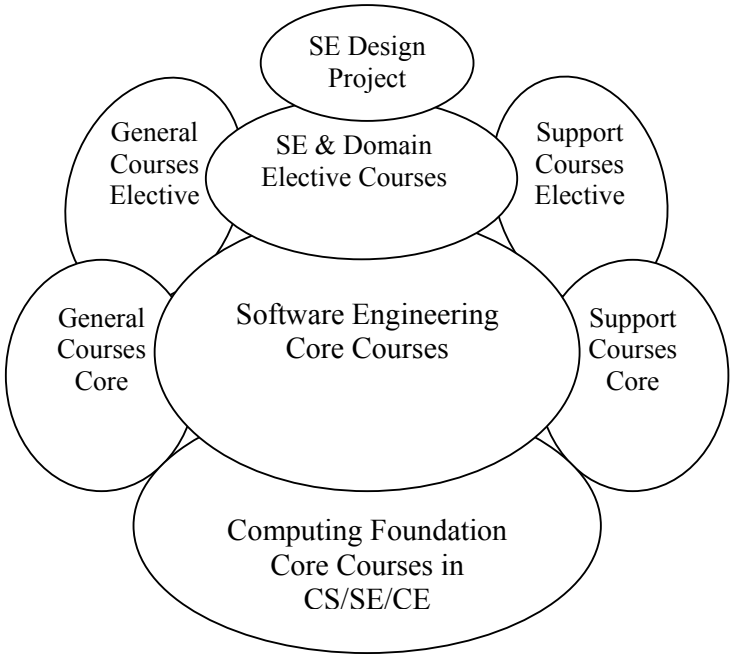



*Figure-1*

| Major Areas | Core/ Required | Electives | Credit Hours |
|---|---|---|---|
| **Computing Foundation** | 37 | 15 | 76 (59%) |
| **Software Engineering** | 18 | | |
| **Software Engineering (Application Domain)** | -- | 06 | |
| **Supporting Studies (Math/Science )** | 12 | 9 | 21 (16%) |
| **General Education** | 15 | 18 | 33 (25%) |
| **Total** | **82 (63%)** | **48 (37%)** | **130** |

## Bachelor of Science in Software Engineering: BS (SE)

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

| Required Software Engineering Courses (18/130) | | | | | |
|---|---|---|---|---|---|
| # | Code | Pre-req | Course Title | Credit hours | Semester |
| 12 | SE | 10 | Software Construction | 3 (3-0) | 4 |
| 13 | SE | 10 | Human Computer Interaction (An SE Approach) | 3 (3-0) | 4 |
| 14 | SE | 12 | Software Design and Architecture | 3 (3-0) | 5 |
| 15 | SE | 10 | Software Quality Assurance | 3 (3-0) | 6 |
| 16 | SE | 10 | Software Requirements Engineering | 3 (3-0) | 6 |
| 17 | SE | 15, 16 | Software Project Management | 3 (3-0) | 7 |

| Elective Computing & Software Engineering Courses (15/130)<br>(The list below is by no means exhaustive. Institutions may add new course) | | | | | |
|---|---|---|---|---|---|
| # | Code | Pre-req | Course Title | Credit hours | Semester |
| 18 | CS | - | Discrete Structures – II | 3 (3-0) | 2,3 |
| 19 | CS | 5 | Automata Theory and Formal Languages | 3 (3-0) | 5, 6 |
| 20 | SE | 5 or 6 | Formal Methods | 3 (3-0) | 6-7 |
| 21 | SE | 10 | Software Development Technologies | 3 (2-3) | 6,7 |
| 22 | SE | 10 | PSP and TSP | 3 (3-0) | 7-8 |
| 23 | SE | 10 | Distributed Computing | 3 (3-0) | 7-8 |
| 24 | CE | 8 | Microprocessor Interfacing | 3 (3-0) | 7 |
| 25 | CS | 4, 5 | Analysis of Algorithms | 3 (3-0) | 6 |
| 26 | CS | 2 | Artificial Intelligence | 3 (3-0) | 6 |
| 27 | CE | 8 | Advance Computer Architecture | 3 (3-0) | 4, 5 |
| 28 | CS | 2, 3 | Principles of Programming Languages | 3 (3-0) | 7 |
| 29 | CS | 41 | Computer Graphics | 3 (3-0) | 6 |
| 30 | CS | 26 | Artificial Neural Networks | 3 (3-0) | 8 |
| 31 | CS | 9 | Advance Database Management Systems | 3 (3-0) | 8 |
| 32 | CS | 8 | Data Security and Encryption | 3 (3-0) | 6 |
| 33 | CS | 8 | Real-time systems | 3 (3-0) | 6-7 |
| 34 | CS | 49 | Bio-Informatics | 3 (3-0) | 5-8 |
| 35 | CS | 9 | Data Warehousing and Data Mining | 3 (3,0) | 6, 7 |

## Domain Specific Elective Courses (6/130)

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<br>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<br>Depth in business administration | 6 | 5-8 |
| 5 | IS | Financial and E-commerce Systems | Accounting; Finance<br>Depth in security | 6 | 5-8 |
| 6 | CE | Fault Tolerant and Survivable Systems | Knowledge and skills in heterogeneous, distributed systems;<br>Depth in security, Intrusion detection<br>Failure analysis and recovery | 6 | 5-8 |
| 7 | CE | Safety Critical Systems | Depth in formal methods, Proof of correctness, etc.<br>Knowledge of control systems | 6 | 5-8 |
| 8 | CE | Embedded & Real time Systems | Hardware for embedded systems<br>Languages and tools for development<br>Depth in timing issues; Hardware verification | 6 | 5-8 |
| 9 | BI | Bio-medical Systems | Biology and related sciences<br>Related safety critical systems knowledge | 6 | 5-8 |

| | | | | | |
|---|---|---|---|---|---|
| 10 | SS | Scientific Systems | Depth in related sciences; Depth in statistics<br>Visualization and graphics | 6 | 5-8 |
| 11 | TE | Telecommunication Systems | Depth in signals, information theory, etc.<br>Telephony and telecommunication protocols | 6 | 5-8 |
| 12 | AS | Avionic & Vehicular Systems | Mechanical engineering concepts<br>Related safety critical systems knowledge<br>Related embedded and real-time systems knowledge | 6 | 5-8 |
| 14 | IE | Industrial Process Systems | Control systems<br>Industrial engineering and other relevant areas<br>Related embedded and real-time systems knowledge | 6 | 5-8 |
| 15 | ES | Multimedia, game, and entertainment Systems | Visualization, haptics, and graphics<br>Depth in human computer interface design<br>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<br>Related embedded and real-time systems knowledge<br>Related telecom systems knowledge | 6 | 5-8 |
| 17 | AI | Agent based Systems | Machine learning, Fuzzy logic<br>Knowledge engineering | 6 | 5-8 |

**Computing Requirements-Supporting Sciences 12 Credit Hours (Refer to Computing part)**

| **Elective Supporting Courses (9/130)**<br>*(The list below is by no means exhaustive. Institutions may add new course)* | | | | | |
|---|---|---|---|---|---|
| 5 | MT | 1 | Multivariate Calculus | 3 (3-0) | 2 |
| 6 | MT | 1 | Differential Equations | 3 (3-0) | 4-5 |
| 7 | MT | 3 | Numerical and Symbolic Computing | 3 (3-0) | 5-6 |
| 8 | MT | 3 | Stochastic Processes | 3 (3-0) | 6-7 |
| 9 | Sc | - | Physics-II (Mechanics) | 3(3-0) | 2 |
| 10 | Sc | - | Bio-Chemistry | 3 (3-0) | 4 |
| 11 | Sc | - | Biology | 3 (3-0) | 4 |
| 12 | EE | 4 | Digital Electronics | 4 (3-3) | 3-4 |
| 13 | Sc | -- | Software Engineering Economics | 3(3,3) | 3-4 |

**Computing Requirements-General Education 12 Credit Hours (Refer to Computing part)**

| **Elective General Education Courses (18/130)**<br>*(The list below is by no means exhaustive. Institutions may add new course)* | | | | | |
|---|---|---|---|---|---|
| 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** | **Cr. Hrs.** |
|---|---|---|---|
| Introduction to Computing | 3 | Discrete Structures-I | 3 |
| Programming Fundamentals | 4 | Object Oriented Paradigm | 3 |
| Calculus and Analytical Geometry | 3 | Math Elective (Multivariable Calculus) | 3 |
| Physics I (Electronics) | 3 | Science Elective (Physics II) | 3 |
| English Comprehension & Composition | 3 | Group Dynamics and Communication | 3 |
| | 16 | | 15 |

| **Semester 3** | **Cr. Hrs.** | **Semester 4** | **Cr. Hrs.** |
|---|---|---|---|
| CS/SE Elective I (Discrete Structures-II) | 3 | Operating Systems | 3 |
| Data Structures and Algorithms | 3 | Software Construction | 3 |
| Introduction to Software Development | 3 | Human Computer Interaction | 3 |
| Linear Algebra and Applications | 3 | Science Elective (Engineering Economics) | 3 |
| Pakistan Studies and Islamic Studies | 3 | Digital Logic and Computer Architecture | 3 |
| | | Technical Writing for SE | 3 |
| | 15 | | 18 |

| **Semester 5** | **Cr. Hrs.** | **Semester 6** | **Cr. Hrs.** |
|---|---|---|---|
| Computer Communication and Networks | 3 | Databases | 3 |
| Software Design & Architecture | 3 | SQA and Process Mgt. | 3 |
| Probability and Statistics for Computing | 3 | Software Requirement Engineering | 3 |
| Professional Software Engg. Practice | 3 | Math/Science Elective | 3 |
| | | SE Application Domain | |
| Gen. Edu. Elective I | 3 | Elective –I | 3 |
| Gen. Edu. Elective II | 3 | Gen. Edu. Elective III | 3 |
| | 18 | | 18 |

| Semester 7 | Cr. Hrs. | Semester 8 | Cr. Hrs. |
|---|---|---|---|
| Software Engineering Project I | 3 | Software Engineering Project II | 3 |
| Software Project Management | 3 | CS/SE Elective III | 3 |
| CS/SE Elective II | 3 | CS/SE Elective IV | 3 |
| SE Application Domain Elective -II | 3 | CS/SE Elective V | 3 |
| Gen. Edu. Elective IV | 3 | Gen. Edu. Elective V | 3 |
| | 15 | | 15 |

**BS (SE) − Software Engineering Courses (Required)**

| **Course Name:** Software Construction | |
|---|---|
| **Course Structure:** Lectures: 3 / Labs: 3 | **Credit Hours:** 4 |
| **Prerequisites:** Data Structures | |

**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

*Sample labs and assignments:*

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

**Reference Material:**

Software Engineering by Roger S. Pressman.

| **Course Name:** Human Computer Interaction (An SE Approach) | |
|---|---|
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Introduction to Software Construction | |

**Objectives:** Psychological principles of human-computer interaction. Evaluation of user interfaces. Usability engineering. Task analysis, user-centred design, and prototyping. Conceptual models and metaphors. Software design rationale. Design of windows, menus, and commands. Voice and natural language I/O. Response time and feedback. Colour, icons, and sound. Internationalization and localization. User interface architectures and APIs. Case

studies and project.

*Learning objectives:*

- Upon completion of this course, students will have the ability to:
- Evaluate software user interfaces using heuristic evaluation and user observation techniques
- Conduct simple formal experiments to evaluate usability hypotheses.
- Apply user centred design and usability engineering principles as they design a wide variety of software user interfaces

**Outline:**

- Background to human-computer interaction. Underpinnings from psychology and cognitive science
- More background. Evaluation techniques: Heuristic evaluation
- More evaluation techniques: Videotaped user testing; cognitive walkthroughs
- Task analysis. User-centred design
- Usability engineering processes; conducting experiments
- Conceptual models and metaphors
- Designing interfaces: Coding techniques using colour, fonts, sound, animation, etc.
- Designing interfaces: Screen layout, response time, feedback, error messages, etc.
- Designing interfaces for special devices. Use of voice I/O
- Designing interfaces: Internationalization, help systems, etc. User interface software architectures
- Expressing design rationale for user interface design

*Sample labs and assignments:*

- Evaluation of user interfaces using heuristic evaluation
- Evaluation of user interfaces using videotaped observation of users
- Paper prototyping of user interfaces, then discussing design options to arrive at a consensus design
- Writers-workshop for style critiquing of prototypes presented by others
- Implementation of a system with a significant user interface component using a rapid prototyping environment

**Resources:**

*HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science* by John Carroll.

*Usability Engineering: Scenario-Based Development of Human Computer* Interaction by Mary Rosson, John Carroll, Mary Beth Rosson.

**Course Name:** Software Design and Architecture

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

**Prerequisites:** Introduction to Software 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

**Outline:**

- In-depth study of design patterns, building on material learned previously.
- Application of design patterns to several example applications
- In-depth study of middleware architectures including COM, Corba, and .Net
- Extensive case studies of real designs.
- Basics of software metrics; measuring software qualities
- Reengineering and reverse engineering techniques.

*Sample labs and assignments:*

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

**Resources:**

*Software Architecture in Practice* by Len Bass, Paul Clements, Rick Kazman
*Evaluating Software Architectures* by Paul Clements, Rick Kazman, Mark Klein

**Course Name:** Software Quality Assurance

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

**Prerequisites:** Introduction to Software Construction

**Objectives:** 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.

*Learning objectives:*

- Upon completion of this course, students will have the ability to:
- Conduct effective and efficient inspections

- 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
- Assess a software process to evaluate how effective it is at promoting quality

**Outline:**

- Introduction to software quality assurance
- Inspections and reviews
- Principles of software validation
- Software verification
- Software testing
- Specification based test construction techniques
- White-box and grey-box testing
- Control flow oriented test construction techniques
- Data flow oriented test construction techniques
- Cleanroom approach to quality assurance
- Software process certification

*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

*Additional teaching considerations:*

User interface testing with end-users is covered in HCI Course, so it should not be covered here.

However the use of test harnesses that work through the user interface is an appropriate topic.

The reason why testing is to be emphasized so much is not that other techniques are less important, but because many other techniques (e.g., inspections) can more easily be learned on the job, whereas testing material tends to require course-based learning to be mastered properly.

**Resources:**

*CMM In Practice: Processes for Executing Software Project at Infosys* by Jalote, Pankaj.

*Software Testing in the Real World: Improving the Process* by Kit, Edward.

| **Course Name:** Software Requirements Engineering | |
|---|---|
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Introduction to Software Construction | |
| **Objectives:** Understand the role of requirements engineering within the software life cycle. Compare and contrast, and evaluate structured, object-oriented, data-oriented, and formal approaches to requirements modelling. Gather the requirements necessary to develop the specifications, given a "customer" who wants a software system to be developed. Develop an informal requirements specification, given a set of requirements. Model, prototype, and specify requirements for a software system. | |
| **Outline:** Software life-cycle models; requirements modelling and analysis; Software requirements specification; Software requirements elicitation and analysis; Structured methods; object-oriented methods; formal methods in requirements (formal and executable specifications); requirements verification and validation; requirements elicitation (e.g., scripting, development of use cases and interface); software requirements metrics; prototyping user interfaces; customer acceptance of requirements. | |
| **Resources:**<br>Davis, A. *Software Requirements: Objects, Functions and States*, Prentice Hall.<br>Dorfman, M. Thayer, R & Davis, A. Eds. *Software Requirements Engineering*, IEEE –CS**.** | |

| **Course Name:** Software Project Management | |
|---|---|
| **Course Structure:** Lectures:3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Introduction to Software Development | |
| **Objectives:** To develop ability to plan and manage software development projects successfully, maximizing the return from each stage of the software development life cycle. | |
| **Outline:** Software development process and phases. Project Management processes and phases. Project initiating, planning, execution, control, monitoring, and closing. Identification of resource, scheduling, PERT, CPM, estimation, FPA, COCOMO. Managing people, resource management, project selection, budgeting, risk management, software processes and standards, process improvement framework. | |
| **Resources:**<br>*Software Project Management in Practice* by Jalote, Pankaj.<br>*A Project Management Body of Knowledge*, by Project Management Institute (PMI).<br>*A Managers Guide to Software Engineering* by Pressman, Roger S.<br>Walker Royce, *Software Project Management - A Unified Framework.*<br>W.J. Brown, et al, *Antipatterns in Project Management, 2000.*<br>Mike Cotterell and Bob Hughes, Software Project Management (Second Edition)*, McGraw Hill, 1999.*<br>Pankaj Jalote*, Software Project Management in Practice, 2002.*<br>E.M. Bennatan, *On Time Within Budget SPM Practices and Techniques 3rd Edition, 2000.* | |

**BS (SE) − Software Engineering Courses (Electives)**

| | |
|---|---|
| **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. | |
| **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:**<br>*System Development using VDM* by Jones, C. B.<br>*Z – Specification Language* by Spiveny | |

| | |
|---|---|
| **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 | |
| **Outline:** IS Audit charter, Polices, 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:**<br>*Control Objective for Information Technology (COBIT),* 3rd Ed, by Information System Audit and Control Foundation.<br>*CISA Review Manual, 2004*, by Information System Audit and Control Association, www.isaca.org. | |

| | |
|---|---|
| **Course Name:** Design and Architecture of Large Software Systems | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Software Analysis and Design | |
| **Objectives:** Modelling and design of flexible software at the architectural level. Basics of model-driven architecture. Architectural styles and patterns. Middleware and application frameworks. Configurations and configuration management. Product lines. Design using Commercial Off-The-Shelf (COTS) software. | |
| **Outline:** Upon completion of this course, students will have the ability to:<br>• Take requirements for simple systems and develop software architectures and high- level designs<br>• Use configuration management tools effectively, and apply change management processes properly<br>• Design simple distributed software<br>• Design software using COTS components<br>• Apply a wide variety of frameworks and architectures in designing a wide variety of software<br>• Design and implement software using several different middleware technologies<br>*Additional teaching considerations:*<br>Students will be taking this before coverage of low-level design. Students, therefore, need tools and packages that allow them to implement their designs without much concern for low-level details. | |
| **Resources:**<br>Development and Maintenance of Large Scale Software Systems By Anonio Pizzarello (Latest Edition). | |

| | |
|---|---|
| **Course Name:** Distributed Computing (3) | |
| **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 | |
| **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:**<br>*Distributed Systems: Principles and Paradigms* by Tanen Baum. | |

| | |
|---|---|
| **Course Name:** Software Development Technologies | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Software Analysis and Design | |
| **Objectives:** Comparative study of different tools and technologies to be used in each phase of SDLC. | |
| **Outline:** It includes CASE as well as other development tools and formalisms. | |
| **Resources:** Various available URLS | |

| | |
|---|---|
| **Course Name:** Software Testing | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Introduction to Software Construction | |
| **Objectives:** Testing techniques and principles: Defects vs. failures, equivalence classes, boundary testing.<br>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.<br>*Learning objectives:*<br>Upon completion of this course, students will have the ability to:<br>o Analyze requirements to determine appropriate testing strategies.<br>o Design and implement comprehensive test plans<br>o Apply a wide variety of testing techniques in an effective and efficient manner<br>o Compute test coverage and yield according to a variety of criteria<br>o Use statistical techniques to evaluate the defect density and the likelihood of faults.<br>o Conduct reviews and inspections.<br>*Additional teaching considerations:*<br>This course is intended to be 95% testing, with deep coverage of a wide variety of testing techniques.<br>The course should build skill and experience in the student, preferably with production code. | |
| **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:**<br>*Software Testing in the Real World: Improving the Process* by Kit, Edward. | |

**BS (SE) − Computing Courses (Elective)**

| **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:**<br>o Review of previous course<br>o Predicate logic: Universal and existential quantification; modus ponens and modus tollens; limitations of predicate logic<br>o Recurrence relations: Basic formulae; elementary solution techniques<br>o Graphs and trees: Fundamental definitions; simple algorithms; traversal strategies; proof techniques; spanning trees; applications<br>o Matrices: Basic properties; applications<br>o Computational complexity: Order analysis; standard complexity classes<br>o 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<br>o Discrete probability: Finite probability spaces; conditional probability, independence,<br>o Methods of Proof, Mathematical Induction and Recursion, loop invariants, Pigeon whole principle, Trees and Graphs, Optimization and matching. | |
| **Reference Material:**<br>*Discrete Mathematical Structures* by Rosen.<br>*Discrete Mathematics* by Richard Johnsonbaugh. | |

| **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. |
| **Reference Material:**<br>*Introduction to Computer Theory*, by Denial Cohen, John Wiley & Sons, Inc.<br>*Introduction to Automata Theory, Languages and Computation*, by J Hopcraft, D. Ullman.<br>*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. | |
| **Reference Material:**<br>*Introduction to Algorithms* by Thomas H Carmen. | |

| | |
|---|---|
| **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:**<br>*Artificial Intelligence* by Luger, 4th edition Pearson Education. |

| | |
|---|---|
| **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, 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. | |
| **Reference Material:** *Computer Graphics* by F.S. Hill. | |

| | |
|---|---|
| **Course Name:** Distributed Database Systems | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Databases | |
| **Objectives:** The student will learn the design, algorithms and techniques involved in distributed database system and their implementation. | |
| **Course Outline:** Advanced data models. Conceptual Database design. Concurrency control techniques. Recovery techniques. Query processing and optimization. Integrity and security. Client-Server architecture. Distributed database systems. Current trends in database systems. Database machines. | |
| **Reference Material:**<br>*Distributed Databases* by Ceri and Pelagatti. | |

| | |
|---|---|
| **Course Name:** Visual Programming | |
| **Course Structure:** Lectures: 2 / Labs: 3 | **Credit Hours:** 3 |
| **Prerequisites:** Data Structures, Data and Network Security | |
| **Objectives:** To development applications using various tools and APIs in visual programming. | |

| |
|---|
| **Course Outline:** Introduction to Windows programming, Use of Windows API, MFC Class hierarchy, Class Wizard, Application Wizard and Application Studio, Graphics Device Interface, Menus, document view architecture, Multiple Views, files and archiving mechanisms, converting Windows programmes to MFC, Sub-classing controls. |
| **Reference Material:**<br>*MFC from the Ground Up.*<br>*Windows 98 API Programming.*<br>*VC++ A complete References.* |

| | |
|---|---|
| **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, command language 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 O 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 Binaryble format (ELF), Dynamic shared objects, Unix Kernel Programming (Ring O), Unix Device Architecture (Character & Block Devices), Device Driver Development, Enhancing Unix Kernel. | |
| **Reference Material:**<br>*The UNIX Programming Environment* by B. Kernighan & R. Pike.<br>*System Software* by Leland L. Beck. | |

| | |
|---|---|
| **Course Name:** Web Engineering | |
| **Course Structure:** Lectures: Labs: (2-3) | **Credit Hours:** 3 |
| **Prerequisites:** Data Warehousing | |
| **Objectives:** Design and implementation of web based applications. | |
| **Course Outline:** Overview of Protocols: TCP/IP, HTTP, Overview of 3-tier Architecture, Web Based Applications Architecture. Developing Front End Applications: Front End Development Tools, HTML, DHTML, Scripting (Java Script, Jscript, VB script), Java Applets, ActiveX. | |
| **Reference Material:**<br>*Web enabled Commercial Application Development Using… HTML, DHTML, JavaScript, Perl, CGI* by Ivan Bayross. | |

**BS (SE) − Supporting Courses (Elective)**

| | |
|---|---|
| **Course Name:** Multivariable Calculus | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Calculus | |
| **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, Difference Equations | |
| **Reference Material:** *Calculus and Analytical Geometry* by Swokowski, Olinick and Pence. | |

| | |
|---|---|
| **Course Name:** Differential Equations | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Multivariable Calculus | |
| **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:**<br>*Advanced Engineering Mathematics* by Kreyzig. | |

| | |
|---|---|
| **Course Name:** Numerical Computing | |
| **Course Structure:** Lectures: 3/ Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Calculus and Analytical Geometry | |
| **Objectives:** On completion of this unit, students will be able to implement numerical solutions to problems using computer-based techniques. | |
| **Course Outline:** Mathematical Preliminaries, Solution of Equations in one variable, Interpolation and Polynomial Approximation, Numerical Differentiation and Integration, Initial Value Problems for Ordinary Differential Equations, Direct Methods for Solving Linear Systems, Iterative Techniques in Matrix Algebra, Solution of non-linear equations. Approximation Theory. Eigenvalues and Eigenvector computation. Optimization procedures. | |
| **Reference Material:**<br>*Elements of Numerical Analysis* by Dr. Faiz, M. Afzal. | |

| | |
|---|---|
| **Course Name:** Physics-II (Mechanics) | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours: 3** |
| **Prerequisites:** None | |
| **Course Outline:** Review of Motion: position, velocity, and acceleration vectors. *Applications of laws of motion:* Projectile Motion, motion in resistive media, rocket motion, motion of charged particles in electrical and magnetic fields**,** *Rotational motion:* constant angular acceleration, uniform circular motion, torque, linear and angular momentum and their conservation, *System of Particles:* centre of mass, two-body collisions in two-dimensions, moment of inertia of objects, *Wave Motion:* mathematical concepts of simple and damped harmonic motion, analytical treatments of superposition of waves, concepts and applications of diffraction and polarization of light and sound waves. Concepts of thermodynamics. | |
| **Reference Material:**<br>*University Physics* by Freedman and Young (10 th and higher editions).<br>*College Physics* by Resnick, Halliday and Krane (6th and higher edition). | |

| | |
|---|---|
| **Course Name: Basic Electronics** | |
| **Course Structure:** Lectures: 3 Labs: 3 | **Credit Hours:** 4(3+3) |
| **Prerequisites:** Electric Circuits | |
| **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:**<br>*University Physics* by Freedman and Young (10 th and higher editions).<br>*College Physics* by Resnick, Halliday and Krane (6th and higher edition). | |

## BS (SE) − General Education Courses (Elective)

| | |
|---|---|
| **Course Name:** Financial Accounting | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Introduction to Accounting and its concepts. Recording Business Transactions: Journal, Ledger, Trial Balance. Preparation of Financial Statements: Balance Sheet, Income Statement, Completion of Accounting Cycle: Adjustments, Closing, Work Sheet Accounting for purchase and sales of merchandise. Receivable and payable, Inventories, Payroll Systems. Plant and Equipment: Acquisition, Depreciation, Disposal. Corporations: Organization | |

| |
|---|
| and stock-holders equity, Operations, Earning per share and dividends. |
| **Reference Material:**<br>*Accounting: The Basis for Business Decisions* by Meigs & Meigs, 10th Edition. |

| | |
|---|---|
| **Course Name:** Financial Management | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Introduction to Financial Management, Concepts and Models in Valuation, The time value of money, Fundamentals of risk and portfolio analysis, Valuation of stocks and bonds, The capital Asset Pricing Model, the Arbitrage Pricing Model and other valuation models. The Cost of Capital: Capital structure and Dividend Policy, The cost of capital, Capital structure theory, Capital structure policy and optimal capital structure, Internal financing and dividends policy Capital Budgeting: The basis of capital budgeting, The determination and use of cash flow, Mutually exclusive investments and capital rationing, Annual equivalent cost and replacement decisions, Risk analysis and the optimal capital budget, Islamic guidelines for financial management: The rational of prohibition of interest, Alternate capital structure, Capital Budgeting in an Interest free economy, working Capital Management in 100% equity capital structure. | |
| **Reference Material:**<br>*Financial Management* by Charles H. Gibson. | |

| | |
|---|---|
| **Course Name:** Human Resource Management | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** An overview of Human Resource Management and Human Resource Manager. The Environment of Human Resource Management, external and Internal Environment. Equal Employment Opportunity and Affirmative Action. Job Analysis: A Basic Human Resource Tool. Human Resource Planning, Recruitment, and Selection. Organization Change and Human Resource Development. Corporate Culture and Organization Development. Career Planning Development. Performance Appraisal. | |
| **Reference Material:**<br>*Managing Human Resource* by Wayne F. Cascio. | |

| | |
|---|---|
| **Course Name:** Organizational Behaviour | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Fundamentals of Organizational Behaviour, Behavioural Science and Organizational Behaviour, Individual Behaviour in Organizations, Personality, perception and attitudes, Learning and reinforcement, Motivation, Group Behaviour in Organizations, Group dimensions in organizations, Group dynamics, Leadership, Organizational Structure and Organizational Behaviour, Organizational design, Job design, Stress and work, Organizational Processes, | |

| |
|---|
| The decision-making process, The communication process, Performance appraisal process, Special Issues in Organizational Behaviour, Management of conflict and change, Organizational development, Impact of computer technology. |
| **Reference Material:**<br>*Organizational Behaviour* by Fred Luthans. |

# 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. 120 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**

- Two years programme (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* of 6 or *Thesis* of 9 credit hours.

**Core Courses**

Following 4 courses are the core.

| S.No. | Code | Course Title | Cr. Hrs. | Semester |
|---|---|---|---|---|
| **1** | **SE** | Software Requirement Engineering | **3** | **1-2** |
| **2** | **SE** | Software System Design & Architecture | **3** | **1-2** |
| **3** | **SE** | Software Quality Engineering | **3** | **1-2** |
| **4** | **SE** | Software Project Management | **3** | **1-3** |

**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** (University/Department may add courses to the list of Electives.)

| | Elective Courses | | Elective Courses |
|---|---|---|---|
| 1 | Formal Methods | 7 | Software Reliability |
| 2 | Risk Management | 8 | Software Costing and Estimation |
| 3 | Design Patterns | 9 | Business Process Re-engineering |
| 4 | Software Metrics | 10 | Personal Software Process |
| 5 | Software Processes | 11 | Reverse Engineering |
| 6 | Distributed Software Development Management | 12 | Human Factors in Computing |

**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 | Software System Design & Architecture | 3 |
| 2 | SE | Software Requirement Engineering | 3 |
| 3 | SE | Elective–I | 3 |

Total 12 Credit Hrs.

**Semester 2**

| S.No. | Code | Course Title | Cr. Hrs. |
|---|---|---|---|
| 1 | SE | Software Quality Assurance | 3 |
| 2 | SE | Software Project Management | 3 |
| 3 | SE | Elective–II | 3 |

Total 9 Credit Hrs.

**Semester 3**

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

Total 6 Credit Hrs.

**Semester 4**

| S.No. | Code | Course Title | Cr. Hrs. |
|---|---|---|---|
| 1 | SE | Thesis–II<br>(3+6 = 9) | 6 |
| Total | | | 6 Credit Hrs. |

**MS (SE) − Core Courses**

| **Course Name:** Software System Design & Architecture | |
|---|---|
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** In-depth study of commonly used software system structures and techniques for describing, designing, and implementing these structures, models and formal notations for characterizing and reasoning about architectures and its implementation. | |
| **Outline:** *Software Design Paradigms & Software Architectures* (design paradigms, design framework, problem decomposition, software architecture, components, integration, implementation); *Software Architectures* (batch-sequential architecture, pipes and filters, layered architecture, client-server architecture, event broadcasting, repositories); *Design Patterns* (creational patterns, structural patterns, behavioural patterns); Frameworks; Aspect-oriented programming; Architecture evaluation. | |
| **Resources:**<br>*Software Architecture in Practice* by Bass, Len.<br>*Software Architecture in Practice* by Len Bass, Paul Clements, and Rick Kazman, Addison-Wesley, 1998.<br>*Design Patterns: Elements of Reusable Object-Oriented Software,* by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Addison Wesley. October 1994.<br>*AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis* by William J. Brown, Raphael C. Malveau, Hays W. "Skip" McCormick III and Thomas J. Mowbray Wiley. 1998.<br><br>**Additional Reading List**<br>*Applied Software Architecture,* Hofmeister, Nord & Soni, Addison-Wesley, 1999.<br>*Pattern-Oriented Software Architecture,* Volume 1: A System of Patterns, Frank Buschmann, et al.<br>*The Patterns Handbook,* Linda Rising, 1998.<br>*The Software Architect's Profession: An Introduction* by Marc T. Sewell, Laura M. Sewell.<br>*Software Architecture: Perspectives on an Emerging Discipline* by Mary Shaw, Garlan David (Textbook Binding) Prentice Hall, 1996.<br>*Evaluating Software Architectures: Methods and Case Studies*, by Paul Clements, Rick Kazman, and Mark Klein. | |

| **Course Name:** Software Requirement Engineering | |
|---|---|
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** This course is aimed at providing an in-depth study of current research and practices in requirements elicitation, requirements analysis, requirements specifications, requirements verification and validation, and requirements management. | |

**Outline:** The Requirements Elicitation Process: Joint Application Design, Prototyping, Requirements Inspections, Quality Function Deployment, Scenarios. Organizing and Analysing the problem. Software Behaviour Specification: State-oriented, Function-oriented, Object-oriented. Formal Methods. Documentation for Software Requirements Specification. Specifying Non-behavioural Requirements. Refinement of requirements into preliminary design.

**Resources:** *Managing Software Requirements: A Unified Approach* by Leffingwell, Dean, 1st Ed.
*Software Requirements Engineering* by Thayer, Richard H.(ed.), 2nd Edition.
Software Requirements by Karl E. Wiegers, 1999.
Software Requirements Engineering by Thayer, Richard H. (ed.), 2nd Edition.

| **Course Name:** Software Project Management | |
|---|---|
| **Course Structure:** Lectures: 3 | **Credit Hours:** 3 |
| **Prerequisites:** None | |

**Objectives:** In depth study of current and emerging trends in theory and practice of software project management. The course focuses on current practice, research and trends in: project planning, cost estimation and scheduling. Project management tools. Factors influencing productivity and success. Productivity metrics. Analysis of options and risks. Planning for change. Management of expectations. Release and configuration management. Software process standards and process implementation. Software contracts and intellectual property. Approaches to maintenance and long-term software development. Case studies of real industrial projects.

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

- Develop a comprehensive project plan for a significant development effort
- Apply management techniques to projects that follow agile methodologies, as well as methodologies involve larger-scale iterations or releases
- Effectively estimate costs for a project using several different techniques.
- Apply function point measurement techniques
- Measure project progress, productivity and other aspects of the software process
- Apply earned-value analysis techniques
- Perform risk management, dynamically adjusting project plans
- Use configuration management tools effectively, and apply change management processes properly
- Draft and evaluate basic software licenses, contracts, and intellectual property agreements, while recognizing the necessity of involving legal expertise
- Use standards in project management, including ISO 10006 (project management quality) and ISO 12207 (software development process) along with the SEI's CMM and CMMI model

*Suggested Sequence of coverage:*
Basic concepts of project management; Managing requirements; Software

lifecycles; Software estimation; The project plan; Monitoring the project; Risk analysis; Managing quality and People problems.

*Sample labs and assignments:*

- Use a commercial project management tool to assist with all aspects of software project management. This includes creating Gantt, PERT, and Earned Value charts
- Make cost estimates for a small system using a variety of techniques
- Developing a project plan for a significant system
- Writing a configuration management plan
- Using change control and configuration management tools
- Evaluating a software contract or licence
- Risk analysis and management case studies

**Resources:**

Futrell, Shafer, Shafer; "Quality Software Project Management", McConnell, Steve, "Software Project Survival Guide".

Bennatan, E.M., "On Time Within Budget".

DeMarco, Tom, "Peopleware" Brooks, Frederick P., "The Mythical Man-Month".

Kerzner, Harold, "Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 7th Ed." Larman, Craig, "Applying UML and Patterns, 2nd Ed."

Publications of Carnegie Mellon's Software Engineering Institute (SEI) on CMM and other Capability Maturity Models.

*Current research publications and literature, and URLS where such courses are being offered.*

**Course Name:** Software Quality Assurance

| **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.

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

- Conduct effective and efficient inspections
- 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
- Assess a software process to evaluate how effective it is at promoting quality

*Suggested sequence of teaching modules:* Introduction to software quality assurance; Inspections and reviews; Principles of software validation; Software verification; Software testing; Specification based test construction techniques; White-box and grey-box testing; Control flow oriented test construction techniques; Data flow oriented test construction techniques; Cleanroom approach to quality assurance; Software process improvement and certification

*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:**

Boris Beizer, *Software Testing Techniques (second edition).*

Graham, et al, *Software Test Automation : Effective Use of Test Execution Tools*, 1999.

*Current research publications and literature and URLS where such courses are being offered.*

**Elective Courses**

| | |
|---|---|
| **Course Name:** Formal Methods for SE | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** Software specification using a mathematically rigorous approach, formal methods and concepts; formal specification languages; algebraic and model-based specification; formal verification and validation methods; proof of correctness of designs and implementations. | |
| **Outline:** Discussion of formal specification, design, and automatic analysis of software systems. The course will present a variety of specification notations (propositional and predicate logic, Z, B, Alloy, UML/OCL, temporal logic, etc.), and discuss corresponding analysis techniques (theorem proving, constraint checking, animation, model checking) using existing commercial and research tools (Jape, Z/Eves, Alloy, USE, SMV). | |
| **Resources:** *Software Engineering: A Practitioner's Approach* by Pressman, Roger S., 5th Edition. | |

| | |
|---|---|
| **Course Name:** Software Testing and Reliability | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** The objective of this course is to understand the issues involved in software reliability and testing and the factors that affect software reliability and guide software testing; to be familiar with a range of standards, techniques and tools developed to support software testing and the production of highly reliable software systems; and to be able to develop software testing strategies. | |
| **Outline:** Reliability concepts and design techniques, management techniques, reliability models, issues of software security; Formal and informal testing | |

| |
|---|
| methods; programme analysis: dynamic analysis, static analysis, data flow analysis; selection of test cases; programme instrumentation; mutation analysis; and symbolic execution. Formal techniques: proving systems correct, checking consistency and completeness. Techniques for safety critical and secure systems. Trustworthiness vs. reliability. Timing analysis and verification. Safety analysis. Fault tolerant systems. |
| **Resources:**<br>*Software Testing Fundamentals: Methods and Metrics* by Hutcheson, Marnie L.<br>*Software Testing Techniques* by Beizer, Boris, 2nd Edition. |

| | |
|---|---|
| **Course Name:** Advanced Software Engineering | |
| **Course Structure:** Lectures: 3 / Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Objectives:** The intent of the course is to provide expertise dealing with current issues related to the development of complex, large-scale software systems, and to develop analytical skills required to implement software process, tools and techniques. | |
| **Outline:** Consideration and use of engineering principles to design and implement cost-effective, reliable software; Current software requirements methodologies and design practices, documentation standards, software project management, verification and validation techniques, software security considerations and computer human interfaces; Agent based software engineering; Clean-room process model; Management and evolution; costing and estimation; legacy systems**.** | |
| **Resources:**<br>*Applying UML and Patterns* by Larman, Craig, 2nd Edition.<br>*Software Engineering* by Pressman, Roger S., 5th Edition. | |

# Information Technology Curricula 2004

# National Curriculum Revision Committee – Information Technology

A three-day final meeting of the Committee (Information Technology) pertaining to the development of curricula for Information Technology degree programmes (BS, MS and Ph.D) and deliberation of related matters was held at Allama Iqbal Open University under the sponsorship of the Higher Education Commission (HEC), Islamabad. The primary objective of the meeting was to discuss and finalize the curricula drafted by the same Committee in the last meeting held at the HEC. Following experts participated in the meetings:

1. Professor Dr. Farhana Shah **Convener**
   Director, Institute of Information Technology
   Chair, Department of Computer Science
   Quaid-i-Azam University
   Islamabad

2. Dr. Naveed Ikram **Secretary**
   Dean
   Faculty of Information Technology
   Riphah International University
   Islamabad

3. Professor Nazir Ahmed Sangi **Member**
   Dean
   Faculty of Science
   Allama Iqbal Open University
   Islamabad

4. Professor Nazir Ahmed Sangi **Member**
   Dean
   Faculty of Science
   Allama Iqbal Open University
   Islamabad

5. Mrs. Saima Jawad **Member**
   Department of Computer Sciences
   Bahria Institute of Management and Computer Sciences
   Bahria University, Shangrila Road, E-8, Naval Complex
   Islamabad

6. Dr. Qasim Sheikh **Member**
   National University of Computer & Emerging Sciences
   Fast House, Rohtas Road, G-9/4
   Islamabad

7. Professor Jafar-ur Rehman **Member**
   Dean
   Faculty of Engineering & Computer Science
   M.A Jinnah University
   Islamabad

8. Dr. Aftab A. Maroof **Member**
   Director
   National University of Computer &
   Emerging Sciences
   Fast House, Rohtas Road, G-9/4
   Islamabad

9. Dr. Jamil Swar **Member**
   Director
   Barani Institute of Information Technology,
   Murree Road
   Rawalpindi

10. Dr. Muhammad Afzal **Member**
    Director
    Centre for Information Technology
    University of Arid Agriculture
    Rawalpindi

11. Dr. M. Ejaz Sandhu **Member**
    University of Central Punjab
    64-EI, Gulberg III
    Lahore

12. Dr. Abdul Aziz Bhatti **Member**
    President
    ACIT Corpn, 69-A Tipu Block, New Garden Town
    Lahore

13. Professor Abdul Karim Baloch **Member**
    Co-Director, Institute of Information Technology
    Mehran University of Engineering & Technology
    Jamshoro

14. Professor Imdad Ali Ismaili **Member**
    Institute of Information Technology
    University of Sindh
    Jamshoro

15. Mr. Mushtaq Ahmed Korai **Member**
    Incharge/Chairman
    Department of Computer System Engg.
    Quaid-e-Awam University of
    Engineering Sciences & Tech.
    Nawabshah

16. Mr. A. Kabeer Qazi **Member**
    Khadim Ali Shah Institute of Technology
    84 B, SMCHS,
    Karachi

17. Dr. Mohammad Altaf Mukati **Member**
    Associate Professor
    Main Campus
    Hamdard University of Information
    & Technology
    Hamdard University
    Karachi

18. Professor Syed Irfan Hyder **Member**
    VP and Dean
    PAF Karachi Institute of Economics & Technology
    Korangi Creek Base
    Karachi

19. Professor Mahmood Khan Pathan **Member**
    Department of Computer Science &
    Information Technology
    NED University of Engg. & Tech.
    Karachi.

20. Professor S. M. Aqil Burney **Member**
    Department of Computer Science
    University of Karachi,
    Karachi

21. Mr. Zafar Malik **Member**
    Head of Computer Sciences
    Muhammad Ali Jinnah University
    Islamabad

22. Mr. Shahid Sheikh **Member**
    National Development Complex
    Islamabad

23. Professor Jamil Ahmed **Member**
Dean, Iqra University
H-9,
Islamabad

24. Dr. Syed Asadullah Shah **Member**
Dean, Isra University
Hyderabad

25. Mr. Muhmmad Sameer Khan **Member**
Institute of Business, Management and Sciences
NWFP Agricultural University
Peshawar

26. Mr. Muhammad Naeem **Member**
Assistant Professor
Department of Computer Science
University of Peshawar
Peshawar

## INFORMATION TECHNOLOGY

Information Technology (IT) is a relatively new discipline and, thus, requires a careful understanding. Technology changes so rapidly and new opportunities and threats arrive so frequently, that IT professionals need to be equipped to identify new sources of knowledge and to constantly learn by themselves. They apply available technology and are motivated by using the computer as a tool to solve real-world problems for people. The objective is to develop human resources to harness the potential of IT as a key contributor to professional development in a variety of fields. This enables maximization of efficiency and productivity in an economy to position it as a key player in the knowledge-driven globalised world.

IT is distinct from other computing disciplines such as Computer Science and Software Engineering. It enshrine a more humanistic perspective of Computer Science and revolves around a deeper study of critical information technology topics, including: system configuration and administration; computer and network hardware installation and maintenance; use and management of databases; development and modelling; creation and management of websites and web-based systems; e-governance and e-commerce; digital voice and video communications; and computer and information security. The curriculum design should keep in view standards and certification requirements to ensure the employability of graduates. All enterprises, whether governmental, commercial, manufacturing, educational, religious, or charitable need a group of professionals who can set up and maintain computer systems along with networks and Internet/web access facilities.

IT professionals do *not* need to be able to design and wire networks. Instead, they should be able to understand the requirements of top management and analyze enterprise needs and capabilities. They should be able to create and manage websites, implement and teach application software, configure and manage databases, test software, and manage software development projects. They should also be able to purchase and install appropriate off-the-shelf hardware and software. It is their responsibility to train and support users in using application programmes, information systems, and databases. IT professionals are responsible for setting up and maintaining information systems wherever needed by them. The widespread demand for such systems has generated a critical need for skilled individuals to lead the establishment and maintenance of information networks, the protection of systems from unwanted intrusion, and the management and dissemination of information via the World Wide Web.

Today, organizations operate and compete in the networked global economy. Advances in information and telecommunication technologies have created knowledge-centred organizations, which in turn develop high value-added

products and services. In such an environment, educating competent knowledge workers and future corporate leaders is a national priority for Pakistan.

**Outcome of IT Programme:**

*Information Technology* (IT) specialists face high expectations with respect to using technology and to the planning, implementation and configuration of computing infrastructure. IT Graduates would possess the combination of knowledge and hands-on expertise and would be able to:

- select hardware and software products appropriate for an organization
- integrate hardware and software products with organizational needs and infrastructure
- Install, customize and maintain hardware and software products
- Install networks; manage network administration and security; the design of web pages; develop multimedia resources; the oversight of email products; and the planning and management of the technology life-cycle by which an organization's technology is maintained, upgraded, and replaced.

## Curriculum for BS Information Technology

The BS programme, proposed in the last meeting, was discussed thoroughly and compared its structure with the recommendations of various international bodies including IEEE and ACM. The report entitled **"Curricula 2004 of ACM and IEEE Curriculum Task Force"** was mainly discussed in detail. All the participants took active part in general discussion on this item of the agenda. The Committee finally agreed to the curriculum model presented in the following table.

### 4.1 PROGRAMME STRUCTURE

| | **Courses Area** | **Credit Hours** |
|---|---|---|
| **Required** | Computing | 37 |
| | Information Technology | 18 |
| | Supporting | 21 |
| | General Education | 15 |
| **Elective** | Information Technology | 21 |
| | General | 18 |
| **Total** | | **130** |

### 4.2 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 (Information Technology). The House unanimously

recommended the eligibility criteria for admission to BS (Information Technology) as given:

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

## 4.3 REQUIRED COURSES:

| S.No. | Course Title | Credit Hours |
|---|---|---|
| 1 | **Computing (Refer to Computing part)** | **37** |
| 2 | **Information Technology** | 18 |
| 3 | Introduction to IT | 3 |
| 4 | Web Engineering | 3 |
| 5 | Systems Administration | 3 |
| 6 | Network Management and Security | 3 |
| 7 | Human Computer Interaction | 3 |
| 8 | Technology Management | 3 |
| 9 | **Computing-Supporting Sciences (Refer to Computing part)** | **12** |
| 10 | **Information Technology-Supporting Sciences** | |
| 11 | Principles of Management | 3 |
| 12 | Organizational Behaviour | 3 |
| 13 | Information Systems | 3 |
| 14 | **Computing-General Education (Refer to Computing part)** | **15** |

## 4.4 ELECTIVE COURSES

There are 21 credit hours for IT electives and 18 credit hours for university electives. It is strongly recommended that these may be non-technical courses such as Introduction to Sociology to build a well rounded personality of students.

## 4.5 COURSE OUTLINE (Core Courses)

| | |
|---|---|
| **Course Name:** Introduction to IT | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours: 3** |
| **Prerequisites:** None | |
| **Course Outline:** Introduction to Databases, DBMS, Types Of Databases, World Wide Web, Web technologies, web based applications, Introduction to Multimedia, Multimedia technologies, Multimedia Applications, Introduction to Computer graphics, Digital graphics, Animation, graphic technologies, Computer networks, internet and intranet, usage of networks, network technologies, social, legal and ethical issues in IT. | |

| |
|---|
| **Reference Material:**<br>An invitation to computer science, second edition, by G. Michael, Schneder, Judith L. Genrsting, Sarabase.<br>Information System Today by Leonard Jessup, Joseph Valacich.<br>Computers Today by Suresh K. Basandra. |

| | |
|---|---|
| **Course Name:** Web Engineering | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours: 3** |
| **Prerequisites:** None | |
| **Course Outline:** An overview of web design concepts, including usability, accessibility, information design, graphic design in the context of web, introduction to web site technologies, Overview of Protocols: TCP/IP, HTTP, Overview of 3-tier Architecture, Web Based Applications Architecture. Developing Front End Applications: Front End Development Tools, HTML, DHTML, Scripting (Java Script, Jscript, VB script), Java Applets, ActiveX. | |
| **Reference Material:**<br>Web enabled Commercial Application Development Using… HTML, DHTML, JavaScript, Perl, CGI by Ivan Bayross. | |

| | |
|---|---|
| **Course Name: Systems Administration** | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours: 3** |
| **Prerequisites:** None | |
| **Course Outline:** A survey of the tools and techniques used in the administration of computing systems, 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, and the use of advanced scripting to ease system administration tasks. | |
| **Reference Material:**<br>The Ultimate Windows 2000 System Administrator's Guide By Robert Williams, Mark Walla. | |

| | |
|---|---|
| **Course Name:** Network Management and Security | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours: 3** |
| **Prerequisites:** Computer Communication and Network | |
| **Course Outline:** Data Communications and Network Management Overview, Review of Computer Network Technology, Standards, Models, and Language, SNMPv1 Network Management: Organization and Information Models, SNMPv1 Network Management: Communication and Functional Models, SNMPv2, SNMPv3, RMON, Broadband Network Management: ATM Networks, Access Networks, Telecommunications Network Management, Network Management Tools and Systems, Network Management Applications, Web-Based Management, Introduction to Key Network Services, Principles and Practices of Network Security, Security Threats and Methods to Avoid Them, | |

NETWORK SECURITY PRACTICE: Authentication Applications, Electronic Mail Security, IP Security, Web Security, SYSTEM SECURITY, Intruders and Viruses, Firewalls, Introduction to Cryptographic Algorithms, Standard Security Protocols.

**Reference Material:**

Network Management, Principles and Practice by Mani Subramanian, Pearson, 0201357429.

Cryptography and Network Security: Principles and Practice, 3/E, William Stallings, Prentice Hall, ISBN: 0-13-091429-0.

| **Course Name:** Network Management and Security | |
|---|---|
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** Computer Communication and Network | |

**Course Outline:** Background to human-computer interaction. Underpinnings from psychology and cognitive science, Evaluation techniques, Heuristic evaluation, Videotaped user testing; cognitive walkthroughs, Task analysis, User-centred design, Usability engineering processes, conducting experiments, Conceptual models and metaphors, Designing interfaces: Coding techniques using colour, fonts, sound, animation, Screen layout, response time, feedback, error messages, Designing interfaces for special devices, Use of voice I/O, Internationalization, help systems, User interface software architectures, Expressing design rationale for user interface design.

**Reference Material:**

Network Management, Principles and Practice by Mani Subramanian, Pearson, 0201357429.

Cryptography and Network Security: Principles and Practice, 3/E, William Stallings, Prentice Hall, ISBN: 0-13-091429-0.

| **Course Name:** Human Computer Interaction | |
|---|---|
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |

**Course Outline:** Background to human-computer interaction. Underpinnings from psychology and cognitive science, Evaluation techniques, Heuristic evaluation, Videotaped user testing; cognitive walkthroughs, Task analysis, User-centred design, Usability engineering processes, conducting experiments, Conceptual models and metaphors, Designing interfaces: Coding techniques using colour, fonts, sound, animation, Screen layout, response time, feedback, error messages, Designing interfaces for special devices, Use of voice I/O, Internationalization, help systems, User interface software architectures, Expressing design rationale for user interface design.

**Reference Material:**

HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science by John Carroll.

Usability Engineering: Scenario-Based Development of Human Computer Interaction by Mary Rosson, John Carroll, Mary Beth Rosson.

| | |
|---|---|
| **Course Name:** Technology Management | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Introduction to Technology Strategy, Corporate Strategy, Technology Transfer, Technology Strategy Development, Product Development Strategy & the Innovation Process. | |
| **Reference Material:**<br>The Ultimate Windows 2000 System Administrator's Guide By Robert Williams, Mark Walla. | |

| | |
|---|---|
| **Course Name:** Principles of Management | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Managers & Management, The Evolution of Management, Managerial Environment, Decision Making, Planning, Strategic Management, Organizing, Human Resource Management, Motivation, Leading, Controlling, Quality, Productivity and Customer Satisfaction and Case Studies. | |
| **Reference Material:**<br>Management by Robins Stephen.<br>Principle of Management by Griffen. | |

| | |
|---|---|
| **Course Name:** Organizational Behaviour | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Fundamentals of Organizational Behaviour, Behavioural Science and Organizational Behaviour, Individual Behaviour in Organizations, Personality, perception and attitudes, Learning and reinforcement, Motivation, Group Behaviour in Organizations, Group dimensions in organizations, Group dynamics, Leadership, Organizational Structure and Organizational Behaviour, Organizational design, Job design, Stress and work, Organizational Processes, The decision-making process, The communication process, Performance appraisal process, Special Issues in Organizational Behaviour, Management of conflict and change, Organizational development, Impact of computer technology. | |
| **Reference Material:**<br>Organizational Behaviour by Fred Luthans. | |

| | |
|---|---|
| **Course Name**: Information Systems | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Systems theory and concepts; information systems and organizational system; decision support; quality; level of systems: strategic, tactical, and operational; system components and relationships; information systems strategies; roles of information and information technology; roles of | |

| |
|---|
| people using, developing, and managing systems; IS planning and change management; human-computer interface; IS development process; evaluation of system performance; societal and ethical issues related to IS design and use. |
| **Reference Material:**<br>Information Systems Development, Paul Lewis, Pitman Publishing. |

## Curriculum for MS Information Technology

The participants were of the opinion that MS (Information Technology) programme should be designed in such a manner that it is the second degree in the field of Information Technology. The following main areas were discussed:

**Eligibility**: BS(CS), BS(SE), BS(IT), MCS, MSc.

1. BS (SE/CS/IT) 4 years degree programme (minimum 130 credit hours) **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 (IT) programme. Candidates will be required to take additional courses to complete credit hour requirement of min. 120 before being formally enrolled in the MS (IT) programme
4. Under eligibility criteria, 1-3 the faculty may recommend additional deficiency courses, from the BS (IT) curriculum, considering the deficiency of the candidates.

## Structure of MS (Information Technology)

- Minimum credit hours shall be 30 for MS (Information Technology) programme.
- The programme shall comprise 4 semesters spread over 2 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:

| **Category or Area** | **Credit Hours** |
|---|---|
| Core | 12 |
| Elective | 12 |
| Thesis | 6-9 |
| **Total Credit Hours** | **30-33** |

## CORE COURSES

| S. No. | Course Title | Crt. Hrs. |
|---|---|---|
| 1 | Advanced Database Systems | 3 |
| 2 | Advanced Computer Networks | 3 |
| 3 | Advanced Web Technologies | 3 |
| 4 | Advanced Multimedia Systems | 3 |

## COURSE OUTLINE

| | |
|---|---|
| **Course Name:** Advanced Databases | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Review of relational databases SQL in the real world: embedded SQL, data passing, status, cursor, connection, transaction, stored procedure; dynamic SQL, parameter, descriptor; JDBC; SQLJ; ODBC. Relational calculus: Object databases, Object-relational databases, objects in SQL:1999. ODMG standard, CORBA: IDL, ORB, dynamic invocation, DB services XML databases, description and query of semi-structured, nested, complex data; XML basics, XML Schema, XSLT, stylesheet, templates, evaluation. XQuery: FLWR expression, evaluation, built-in functions, user-defined functions, aggregation, quantification. More XQuery: data and types; XQuery and XML Schema; proj, sel, construction, group, join, recursive function, wildcard types, XqueryX; XPath and XQuery, laws. Query processing, Query optimization, OLAP, vs OLTP, Vs data mining; multidimensional model, star schema; aggregation, drilling, rolling, slicing, dicing; CUBE, ROLLUP. Materialized views, ROLAP and MOLAP; data mining, associations, priori algorithm, other kinds, machine learning; data warehouse, ETL tools, metadata, incremental updates. | |
| **Reference Material:**<br>Database and Transaction Processing: An Application-Oriented Approach by Philip M. Lewis, Arthur Bernstein, and Michael Kifer. Addison Wesley, 2002. | |

| | |
|---|---|
| **Course Name:** Advanced Computer Networks | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Internet history and architecture, OSI layering, The end-to-end principle, MAC and LLC Issues: Techniques for multiple access, Adaptive LLC mechanisms for wireless link, Internet Routing Architecture: Internet Service Providers and Peering, Border Gateway Protocol (BGP), BGP instability, Fair queuing, TCP congestion control, TCP RTT estimation, Fast retransmit, Fast recovery, Integrated Services, Resource ReSerVation Protocol (RSVP), Differentiated Services, Wireless TCP, Mobile IP, Multicast routing, | |

| |
|---|
| Scalable Multicast routing, Core Based Trees (CBT), Protocol Independent Multicast (PIM), Scalable Reliable Multicast, Overlay Networks, Basics in Cryptography and Security, Distributed Denial of Service, IP Traceback, Domain Name System (DNS), Using DNS for System Break-ins, Simulation and modelling issues, Wide-Area Traffic Modelling, How to evaluate TCP?, Traffic Engineering, Multi-Protocol Label Switching (MPLS), IP Next generation, IPv6, IP Next Layer (IPNL). |
| **Reference Material:**<br>High Performance TCP/IP Networking, Concepts, Issues and Solutions, Hassan and Jain, Prentice Hall.<br>High-Speed Networks and Internets, Stallings, Prentice Hall, 2002. |

| | |
|---|---|
| **Course Name:** Advanced Web Technologies | |
| **Course Structure:** Lectures: 3/Labs: 0 | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** The course is dynamic, accordingly, URLS may be referred. | |
| **Reference Material:**<br>URLS. | |

| | |
|---|---|
| **Course Name:** Advanced Multimedia Technologies | |
| **Course Structure:** Lectures: Labs: | **Credit Hours:** 3 |
| **Prerequisites:** None | |
| **Course Outline:** Introduction to Multimedia Programming, Scope of Multimedia Programming, convention and trends, Media types used in current applications (including digital video, audio, and graphics). System level issues of performance synchronization, storage and server schemes, dynamic interactivity, hyper linking, multimedia device control, distributed media development and delivery, non-standard media and programming frame works. Introduction to Multi-media Networks. | |
| **Reference Material:**<br>*Multimedia Systems Design,*1/e, Andleigh, P.K. and Thakrar, K., Prentice Hall. | |

**List of Suggested Elective Courses and areas of Specialization**

The Committee suggested the following list of electives, however university may add other relevant courses to it.

| Course Title | Credit hours | Course Title | Credit hours |
|---|---|---|---|
| Information Management | 3 | Electronic Coaching Systems | 3 |
| Interactive Media | 3 | Communication Design for the World Wide Web | 3 |
| Enterprise Architecture | 3 | IT Entrepreneurship | 3 |

| | | | |
|---|---|---|---|
| Systems for Teaching and Learning | 3 | Strategic Information Technology Management | 3 |
| Secure Applications/Application Security | 3 | Wireless Technologies and Applications | 3 |
| Strategic Information Systems | 3 | Client Server Technologies and Applications | 3 |
| Multimedia Information Systems | 3 | Digital Voice Communications | 3 |
| Inter-Enterprise Computing | 3 | Project Management for IT | 3 |
| Information Technology and Strategic Opportunity | 3 | Broadband Networks | 3 |
| Internet Protocols | 3 | Management of Technical Projects | 3 |

**Areas of Specialization:**

The following areas of specialization in the IT discipline are suggested:

Data Modelling
Decision Support Systems
Digital Communications
Digital Systems Design
Distributed Systems
Electronic Commerce
Executive Information Systems
Object-Oriented Systems
Robotics
Software Metrics
Formal Specification
Graphics And Image Processing
Human-Machine Interfaces
Inductive Inference
Information Storage And Retrieval
Information Systems Management
Inter-Organizational Systems
Knowledge-Based Systems
Librarianship

# PhD Computing
## (Computer Science, Software Engineering, Information Technology) Curricula 2004

## Ph.D. (Computing: Computer Science, Software Engineering and Information Technology) Programme

The participants actively discussed various aspects of the proposed models for PhD programme suggested by relevant committees (i.e. CS, SE, and IT Curriculum Committees) and propose a combined model for PhD in all disciplines of computing. The detail of Ph.D. programme is available in the following paragraphs.

1. **Eligibility**
    - MS (Computer Science, Software Engineering, Information Technology) 18 years degree.
    - MS degree in related discipline, however such candidates must complete any pre-requisite.
    - BS (CS/SE/IT) 16 years degree or equivalent (e.g. M.Sc., MCS, etc.) Students will be completing coursework prescribed for MS (CS, SE, IT) programme as well.

2. **Evaluation of Candidate**

The respective university may evaluate the eligibility of candidates for entrance into the programme as per their procedure/rule as prescribed by the relevant academic council or board of studies.

3. **Duration**
    - Minimum 3 years after MS (Computer Science)
    - Minimum 4 years after BS (Computer Science)

4. **Structure of Ph.D. (Computer Science)**

**a) After MS … 48 credit hours**

The Ph.D. programme is structured on the basis of minimum of 48 credit hours. The programme requirements involve minimum 12 credit hours course work and 36 credit hours research work.

Furthermore, it is proposed that the Ph.D. course work credits may be implemented via selection of a particular mode of course execution (as recommended by the respective advisor) from the various available approaches including guided, taught, seminars, and independent research studies.

**b) After BS …69 credit hours**

The Ph.D. programme shall comprise minimum of 69 credit hours after BS. The programme requirements may involve minimum 21 credit hours of MS course work, followed by minimum 12 credit hours Ph.D. course work, and minimum 36 credit hours of research work.

**5. Qualifying Evaluation for Ph.D.**

The student shall be evaluated for qualifying for the candidacy of research either during MS course work or after the completion of the MS course work. The procedure of qualifying evaluation may be defined by the respective university.

**6. Research Proposal**

After qualifying the candidacy of research and on completion of MS course work, the student will be allowed to start research part of the Ph. D programme.

**7. Publications of Research Work during Ph.D.**

The participants of the meeting discussed the possibility of introducing of mandatory condition regarding publication of research contributions of the respective students in the reputed journals or international refereed conferences during his/her Ph.D. programme. However, due to various practical constraints involving travelling abroad and availability of budget for the said purpose, it was unanimously agreed that the publications are desirable but may not be mandatory for the submission of Ph.D. thesis.

**8. Thesis Evaluation**

The Ph.D. thesis shall be evaluated by the experts in the area of the respective research. As per HEC guidelines at least two evaluators shall be from universities of international repute.

**9. Thesis Defence**

The Ph.D. student will be required to defend his/her work after positive evaluation by foreign experts. The process of thesis defence may engross constitution of committee including faculty members (experts in the domain of research work) from local Pakistani universities. The degree shall be awarded on successful defence of the particular thesis research work.

**10. General Recommendations**

There were general recommendations by the participants as given below:

I. HEC may encourage and facilitate a research team comprising of multiple universities/institutions. Special arrangements may be made to utilize Ph.D. expertise available at public and private sectors universities for course delivery and co-supervision of thesis.
II. Model collaboration/sharing of facilities proposal shall be developed and funded by HEC.