SOLAPUR UNIVERSITY, SOLAPUR

FACULTY OF ENGINEERING & TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

Syllabus Structure and detailed syllabus of

T.E. (Computer Science & Engineering)

w.e.f. Academic Year 2016-17
Structure of T. E. (Computer Science & Engineering) w. e. f. 2016-17

<table>
<thead>
<tr>
<th>Theory Course Name</th>
<th>Hrs./week</th>
<th>Credits</th>
<th>Examination Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>Operating System Concepts</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>System Programming</td>
<td>3</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Computer Networks</td>
<td>4</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Design and Analysis of Algorithm</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Computer Organization</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Self Learning (HSS)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Sub Total</strong></td>
<td>16</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Laboratory/Workshop</th>
<th></th>
<th></th>
<th>ESE</th>
<th>POE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System Concepts</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Computer Networks</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td># Lab – Java Programming</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td><strong>Sub Total</strong></td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td>18</td>
<td>2</td>
<td>10</td>
<td>27</td>
</tr>
</tbody>
</table>

Abbreviations: L- Lectures, P-Practical, T- Tutorial, ISE- In Semester Exam., ESE - End Semester Exam, ICA- Internal Continuous Assessment
ISE- Internal Tests, ESE - University Examination (Theory & POE & Oral examination)
Note: 1) ‘#’ indicates Practical exam only.
<table>
<thead>
<tr>
<th>Theory Course Name</th>
<th>Hrs./week</th>
<th>Credits</th>
<th>Examination Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>Compiler Construction</td>
<td>3</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Unix Operating System</td>
<td>3</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Mobile Computing</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Database Engineering</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Software Engineering</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Self Learning Module-II (HSS/Technical)</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td><strong>Sub Total</strong></td>
<td>16</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

| Laboratory/Workshop                     |          |        | ESE | POE | OE |
|                                        |          |        |     |     |    |
| Database Engineering                   |          |        | 2   | 1   | 50 |
| Lab - Programming in C# .net           | 2         | 2      | 3   | 50  | 25 |
| Mini project                           |          |        | 2   | 1   | 25 |
| **Sub Total**                          | 2         | 6      | 5   | 125 | 75 |
| **Grand Total**                        | 18        | 10     | 25  | 525 | 175 |

<table>
<thead>
<tr>
<th>Abbreviations: L- Lectures, P –Practical, T- Tutorial, ISE- In Semester Exam., ESE - End Semester Exam, ICA- Internal Continuous Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note: 1) 'P' indicates Practical exam only.</td>
</tr>
</tbody>
</table>
Subjects for Self Learning for Humanities and Social Sciences (HSS)
1. Economics
2. Psychology
3. Philosophy
4. Sociology
5. Humanities

Subjects for Self Learning for Technical Subjects
1. Computer Modeling and Simulation
2. Software licenses and practices
3. Network set up & management tools

Note:
1. The term-work will be assessed based on continuous internal evaluation including class tests, assignments, performance in laboratories, interaction in class, quizzes and group discussions as applicable.
2. The batch size for practical/tutorials be of 15 students. On forming the batches, if the strength of remaining students exceeds 7 students, then a new batch may be formed.
3. Mini Project shall consist of developing small software based on tools & technologies learnt in SE and TE.
5. Vocational Training (evaluated at B.E. Part-I) of minimum 15 days shall be completed in any vacation after S.E. Part-II but before B.E. Part-I & the report shall be submitted and evaluated in B.E. Part-I.
6. Student shall select one Self Learning Module from HSS at T.E. Part I and one self learning module either from HSS or from technical at T.E. Part II.
7. Curriculum for Humanities and Social Sciences Self Learning Modules is common for all under graduate programmes under faculty of Engineering and Technology.

Abbreviations: L- Lecture, P- Practical, T- Tutorial, ISE- In Semester Exam., ESE - End Semester Exam., ICA - Internal Continuous Assessment
ISE - Internal Tests, ESE - University Examination (Theory & POE & Oral examination)
Note: 1) 'P' indicates Practical exam only.
1. OPERATING SYSTEM CONCEPTS

Teaching Scheme
Lecture: 3 Hrs/Week
Practical: 2 Hrs/Week
Tutorial: 1 Hr/Week

Examination Scheme
Theory: 100 Marks
Termwork: 25 marks
Practical Oral Examination: 50 marks

COURSE OBJECTIVES:
1) To expose the importance of the role and structure of operating system.
2) To learn basics of operating system such as Process Management, Memory Management and I/O device management.

COURSE OUTCOMES:
At the end of the course, student will be able to
1) Recognize the role, structure of OS, applications and relationship between them.
2) Analyze the features and functions provided by Operating system modules (such as process control, CPU scheduling, mutual exclusion, deadlock, memory management, synchronization etc.)

SECTION – I

Unit 1: Introduction (5 Hrs.)

Unit 2: Process (6 Hrs.)
Process Concept, Process Scheduling, Operation on process, Cooperating process, Threads, Interprocess Communication.

Unit 3: Process Scheduling (6 Hrs.)
Basic concept, Scheduling Criteria, Scheduling Algorithms, Multiple processor scheduling, Real time scheduling (Algorithms evaluation).

Unit 4: Interprocess synchronization (5 Hrs.)
Background, The critical section problem, Synchronization Hardware, Semaphores, Classical problems of synchronization, Monitors.

SECTION – II

Unit 5: Deadlocks (7 Hrs.)
System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock, combined approach to deadlock.

Unit 6: Memory Management (6 Hrs.)
Background, Logical Versus Physical Address space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with paging.
Unit 7: Virtual Memory (5 Hrs.)
Background, Demand paging, Page replacement, Page replacement algorithms, Allocation of frames, thrashing (Only concept).

Unit 8: I/O system (4 Hrs.)
Overview, I/O hardware, Application I/O interface, Kernel I/O subsystem, Transforming I/O request to hardware operation.

Text Books:

Reference Books:
2. Operating system with applications in UNIX, Novell, and Windows NT by Achyut Godbole (TMGH).

Term work:
Tutorials:
In tutorial session, students will be assigned different exercise problems and be guided for the solution.

Tutorials are based on:
1. Study and Comparison of different types of OS.
2. Examples of IPC such as POSIX shared memory, Mach OS, Windows XP
3. Exercise problems given in textbook for Unit III.
4. Exercise problems given in textbook for Unit VI
5. Exercise problems given in textbook for Unit V.
6. Exercise problems given in textbook for Unit VI.
9. Exercise problems given in textbook for Unit VIII.

Practical List:
It should consist of the 12 practicals based on the list below:
1. Study of UNIX Operating System and required commands.
2. Implementation of a program which describe the use of system calls such as fork (), abort (), suspend () etc.
4. Implementation of SJF (preemptive & non preemptive)
5. Implementation of round robin (RR).
7. Implementation of Banker’s Algorithm for Deadlock Avoidance.
8. Implementation of RAG or WFG method for Deadlock detection for single instance of resources.
10. Implementation of Mutual Exclusion 1st/2nd/3rd algorithm.
11. Implementation of Mutual Exclusion using semaphore (wait & signal)
12. Implementation of producer consumer problem (Bounded buffer)
SOLAPUR UNIVERSITY, SOLAPUR

T.E. (COMPUTER SCIENCE & ENGINEERING)
Semester – I

2. SYSTEM PROGRAMMING

<table>
<thead>
<tr>
<th>Teaching Scheme</th>
<th>Examination Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture: 3 Hrs/Week</td>
<td>Theory: 100 marks</td>
</tr>
<tr>
<td>Practical: 2 Hrs/week</td>
<td>Termwork: 25 marks</td>
</tr>
</tbody>
</table>

**COURSE OBJECTIVES:**
1. To learn the principles of processing of an HLL program for execution on a computer system.
2. To design computer language processors.
3. To acquire skills of Language processor development tools.

**COURSE OUTCOMES:**
At the end of the course, student will be able to
1. Identify various language processors.
2. Design and implement prototypes of language processors.
3. Apply language processor development tools to create Language Processors.

**SECTION – I**

**Unit 1: Language Processors**
(8 Hrs.)

**Unit 2: Assemblers**
(8 Hrs.)
Elements of assembly language programming, A simple assembly scheme, Pass structure of assemblers, design of a two pass assembler, A single pass assembler for IBM PC.

**Unit 3: Macros and Macro Processors**
(6 Hrs.)
Macro definition and call, Macro Expansion, Nested macro calls, Design of Macro preprocessor-Design overview.

**SECTION – II**

**Unit 4: Compilers and Interpreters**
(9 Hrs.)
Aspects of compilation, compilation of expressions, code optimization, Static and dynamic memory allocation, Memory allocation in block structured languages(Scope Rules, Memory allocation and access, Dynamic pointer), Interpreters

**Unit 5: Linkers**
(6 Hrs.)
Relocation and linking concepts, design of a linker, Self-relocating programs, linking for overlays.

**Unit 6: Loaders**
(7 Hrs.)
Function of loader, general loader scheme, Absolute loader, Relocating loader, Direct linking loader, Dynamic loading, Design of direct linking loader.
Text books:
1. System Programming and operating systems – 2nd Edition D.M. Dhamdhere (TMGH) (Unit-1,2,3,4,5)
2. System Programming -- J. J. Donovan (Mc-Graw Hill) (Unit-6)

Reference books:

Termwork:
Practical List:
Practical assignments should be carried based on –
1. Simulation of 
2. Introduction o
3. Implementatic
4. Implementatic
5. Design and im
6. Design and im
7. Symbol table ;
8. Design Lex sj
   white spaces.
9. Implementatio
10. Simulation of
11. Simulation of
SOLAPUR UNIVERSITY, SOLAPUR
T.E. (COMPUTER SCIENCE & ENGINEERING)
Semester – I
3. COMPUTER NETWORKS

Teaching Scheme
Lecture: 4 Hrs/Week
Practical: 2 Hrs/Week

Examination Scheme
Theory: 100 Marks
Term-Work: 25 Marks
Practical Oral Exam: 50 Marks

COURSE OBJECTIVES:
1) To build the idea of multiple layers in the data communication and the addressing mechanism between the different layers of OSI Reference Model.
2) To introduce the student with client-server paradigm for socket interfaces to discuss the client-server communication using connectionless & connection-oriented services offered by the transport layer protocols.
3) To study the architecture of WWW, HTTP, e-Mail & describe the concepts of hypertext, hypermedia, web clients, web servers and their components to define URL, different Web documents in the application layer.

COURSE OUTCOMES:
At the end of the course, student will be able to
1) To demonstrate the purpose of different layers.
2) To write application layer protocols using services offered by the transport layer protocols such as UDP, TCP & SCTP.
3) To show the functioning of web based mail system and web services working mechanism.

SECTION – I

Unit 1: Overview of TCP/IP Protocol Suite (4 Hrs.)

Unit 2: Transport Layer (14 Hrs.)
UDP: Overview of the OSI Model and the TCP/IP Protocol Suite, UDP: Introduction, User Datagram, UDP Services, UDP Applications, UDP Package,
TCP: TCP Services, TCP Features, Segment, A TCP Connection, State Transition Diagram, Window in TCP, Flow Control, Error Control, Congestion Control, TCP Timers, TCP Package

Unit 3: Client Server Model and Socket Interface (8 Hrs.)
Client Server Paradigm: Server, Client, Concurrency, Concurrency in Clients, Concurrency in Servers, Socket, Byte Ordering Functions, Address Transformation Functions, Memory Management Functions, Socket System Calls, Connectionless Iterative Server, UDP Client Server Programs, Connection-oriented Concurrent Server, TCP Client Server Programs.
SECTION – II
Unit 4: Host Configuration & Domain Name System (8 Hrs.)
Host Configuration: BOOTP Operation, Packet format, DHCP: Introduction, DHCP Operation and Configuration.
Domain Name System: Need for DNS, Name Space, DNS In the Internet, Resolution, DNS Messages, Types of Records, Encapsulation, DDNS.

Unit 5: Remote Login and File Transfer (10 Hrs.)
SSH: Components, Port Forwarding, Format of SSH Packets.

Unit 6: WWW, HTTP and Electronic Mail (8 Hrs.)
World Wide Web and HTTP:

Text Books:
1. TCP/IP Protocol Suite: Behrouz A. Forouzan (Fourth Edition) (Unit 1, 2, 3, 5, 6)
4. Computer Networks: Andrew S. Tanenbaum

Reference Books:
1. Internetworking with TCP/IP Vol III. Client-Server Programming & Applications: Douglas E. Comer
2. Data and Computer Communications: William Stallings
3. Data Communication and Networking: Behrouz A. Forouzan

Term work:
Student should perform 10 to 12 experiments based on the following guidelines and preferably conducted on Unix/Linux platform using C language.
1. Installation of Unix/Linux Operating System.
2. Configuration of Network-Assigning IP Address, Subnet-Mask, Default Gateway, DNS Server Addresses & Testing Basic Connectivity.
8. Implementation of Simple Network Chatting Application.
9. Remote Login: TELNET
   a. Log on to a remote computer from client using TELNET.
b. After logging on executes few commands at remote server from client. For example user wants a server to display a file (hello.txt) on a remote server then he/she types: `cat hello.txt`.

c. Log on to a remote computer from client using TELNET and Putty terminal emulator. After logging on execute few commands. Here Client and Server are on heterogeneous systems, for example client is on windows and server is on Linux.

10. Remote Login: SSH
   a. Log on to a remote computer from client using SSH.
   b. After logging on executes few commands at remote server from client. For example user wants a server to display a file (hello.txt) on a remote server then he/she types: `cat hello.txt`.
   c. Log on to a remote computer from client using SSH and Putty terminal emulator. After logging on execute few commands. Here Client and Server are on heterogeneous systems for example client is on Windows and server is on Linux.
   d. Execute a command on remote computer (Linux/Unix) using SSH without logging on. For example to execute a single command on remote computer one may use `ssh user1@server1 command1`.

11. File Transfer: FTP
   a. Connect to a FTP server from client.
   b. Download a file from server. For example, suppose there is a file (hello.txt) on the server then download it on the client.
   c. Upload a file onto server. For example, suppose there is a file (hello.txt) on the client then upload it on the server.
   d. List directory contents. For example, use `ls` command to list the contents of current directory.

12. Simulation of DHCP.

13. Simulation of DNS.

SOLAPUR UNIVERSITY, SOLAPUR
T.E. (Computer Science & Engineering)
Semester - I

4. DESIGN AND ANALYSIS OF ALGORITHM

Teaching Scheme
Lecture: 3 Hrs/week
Tutorial: 1 Hr/week

Examination Scheme
Theory: 100 marks
Term work: 25 marks

COURSE OBJECTIVES:
1. To study algorithm analysis, design and application.
2. To evaluate and compare algorithms using worst, average and best case analysis.
3. To equip the student with essential algorithm design techniques such as divide and conquer, dynamic programming and the greedy methods and many of its applications.
4. To explain the difference between tractable and intractable problems and identify basic complexity classes such as P, NP complete and NP-hard.

COURSE OUTCOMES:
At the end of the course, student will be able to
1. Analyze the asymptotic performance of algorithms.
2. Demonstrate a familiarity with major algorithms.
3. Apply important algorithmic design paradigms and methods of analysis.
4. Synthesize efficient algorithms in common engineering design situations.

SECTION – I (8 Hrs.)

Unit 1: Introduction
Algorithm Specification: Pseudo code Conventions, Recursive Algorithm
Performance Analysis: Space Complexity, Time Complexity, Calculating worst case and average case complexities: Case study, Amortized Complexity, Asymptotic Notations, Practical Complexities, Performance Measurement

Unit 2: Divide and Conquer (7 Hrs.)
The general method, Binary search, Finding the maximum and minimum, Merge sort, Quicksort, Selection, Strassens Matrix multiplications

Unit 3: The Greedy method (8 Hrs.)
The general method, Knapsack problem, Job sequencing with deadlines, minimum-cost spanning trees – Prim’s and Kruskal’s Algorithms, Optimal storage on tapes, Optimal merge patterns, Single source shortest paths

SECTION - II (8 Hrs.)

Unit 4: Dynamic Programming
The general method, Multistage graphs, All pair shortest paths, Optimal binary search trees, 0/1 Knapsack, Reliability design, The Traveling Sales person problem. Flow shop scheduling

Unit 5: Backtracking (7 Hrs.)
The general method, 8-queen problem, sum of subsets, Knapsack Problem, Hamilton Cycle, and Graph Coloring.
Unit 6: NP-Hard and NP-Complete problems (7 Hrs.)
Tractable and Intractable Problems: Computability. The Halting problem, Computability classes – P, NP-complete and NP-hard, Cook’s theorem, Standard NP-complete problems Reduction techniques

Text Books:
1. Fundamentals of Computer Algorithms–Horowitz, Sahni & Rajasekaran (Galgota Publications)
3. Data Structures and Algorithms - Aho, Hopfcraft and Ullman (Addison wesley)

Reference books:
1. Fundamental of Algorithm. – Gilles Brassard, Paul Bratley (Pearson Publication)
2. Introduction to Algorithms – Thomas Cormen (Pearson Publication)
3. Introduction to Design and Analysis of Algorithm – By Goodman (McGrawhill)

Termwork:

Assignment List:
1. Assignment on Space complexity & Time Complexity
2. Problems on Asymptotic Notations
3. Performance measurement of various algorithms
4. Assignment on Recurrence Relations
5. Study and solve problems of different sorting algorithms
6. Finding time complexity of searching and sorting algorithms
7. Solve exercise on Greedy Algorithms
8. Assignment on Dynamic programming
9. Assignment on Backtracking
10. Exercise on NP-Hard and NP-Complete problems
SOLAPUR UNIVERSITY, SOLAPUR
T.E. (Computer Science & Engineering)
Semester - I

5. COMPUTER ORGANIZATION

Teaching Scheme
Lecture: - 3 Hrs/Week

Examination Scheme
Theory – 100 Marks
Term-Work – 25 Marks

COURSE OBJECTIVES
1) To learn fundamentals of computer organization.
2) To know the processor level design, memory and I/O organization.
3) To acquire fundamentals of pipelined architecture.

COURSE OUTCOMES
At the end of the course, student will be able to
1) Justify the principles of computer organization.
2) Identify performance of processor, design memory hierarchy and interface I/O devices.
3) Identify parallel architecture.

Course Instruction
Concepts of CPU may be clarified through simulation software tools.

SECTION I

Unit 1: Basic Structure of Computer Hardware (06 Hrs)
Functional Units, Basic operational concepts, Bus Structure, Generation of computers.

Unit 2: Processor Level Design (11 Hrs.)
Instruction format, Instruction types, Bus hierarchical architecture, RISC, CISC, Fixed point arithmetic-Addition, Subtraction, Multiplication (Booth Algorithm ),Fast multiplication, Division(Restoring and Non Restoring Algorithm) Implementation of floating point operation, IEEE floating point standard

Unit 3: Hardwired Control Unit (05 Hrs.)
Hardwired Control Unit; Design Methods (Sequence counter); Multiplier Control Unit (Introduction), (Implementation of Multiplier in each case).

SECTION II

Unit 4: Memory Organization and Design (08 Hrs.)
Virtual memory: Memory Hierarchy, Main memory allocation, Segments & pages, Replacement policies ,High Speed memories-Interleaved memories, Cache, Associative.

Unit 5: Input-Output Organization (05 Hrs.)
Accessing I/O devices, Direct Memory Access, Interrupt Handling, I/O Interfaces, I/O Channels

Unit 6: Parallel Processing and Pipelined Architecture (09 Hrs.)
Uniprocessor and Multiprocessor parallelism; Types of uniprocessor parallelism; Basics of Pipelining & vector processing, Multiprocessor Architecture-tightly coupled & loosely coupled , Linear and Nonlinear pipeline ,Pipeline hazards.
Text Books:
1. Computer Architecture & Organization – J.P.Hayes (MGH) (Chapters:1,2,3,4)
2. Computer Organization - Hamacher and Zaky (MGH)(Chapters:1,5)
3. Advanced Computer Architecture and Parallel Processing- Kai Hwang and Briggs (MGH) (Chapter:6)

Reference Books:
1. Advanced Computer Architecture- Kai Hwang (MGH)
2. Computer Organization and Architecture – Hennessy Patterson (ELSEVIER)
3. Design for Performance-William Stallings (PEARSON)

Termwork:
Assignment List:
1. Discuss the different generations of computer and compare them.
2. Name the different addressing modes, their assembler syntax, addressing functions with an example.
4. Hexadecimal values in single precision format.
5. With an example discuss the division operation with restoring and non-restoring method.
6. Why program control of I/O is unsuitable for high speed data transfer. Give the reasons.
7. Discuss the sequence of events involved in handling an interrupt request from a single device.
8. What are the major difference between Cache-Main memory and Main-Secondary memory hierarchies?
9. With an example, explain different memory capacities.
10. Draw the architecture of tightly coupled multiprocessor with private cache.
11. What is hazard? What are different types of hazards? Explain each with an example. What are types of data hazard? With example, explain types of data hazard.
SOLAPUR UNIVERSITY, SOLAPUR
T.E. (Computer Science & Engineering)
Semester - I
6. Lab - JAVA Programming

Teaching Scheme  Examination Scheme
Lecture:  2 Hrs/Week  Term-Work:  25 Marks
Practical:  4 Hrs/Week  Practical Oral Exam:  50 marks

COURSE OBJECTIVES:
1. To learn Object oriented programming paradigms using Java language.
2. To introduce the student Basic Java API Classes and Features for use in Application programming.
3. To impart basic understanding and analyze platform independent application runtime environment to create standalone GUI, Web applications using Java language.

COURSE OUTCOMES:
At the end of the course, student will be able to
1. Implement Object oriented programming paradigms using Java language.
2. Explore and use the Java APIs for implementing various functionalities of an Application.
3. Analyze platform independent application runtime environment and choose appropriate runtime environment to create GUI and Web applications using Java language.

SECTION I
Unit 1: Basics and Fundamentals of Java (4 hrs)
Fundamentals: Data Types, Arrays, Objects and Classes. Fields and Methods, Access control, Modifiers, Constructors, Overloading methods, Abstract classes, Nested classes, Packages, Wrapper classes, Interfaces, Using the Keyword “this”. Object Life time & Garbage Collection. Recursion in Java.

Unit 2: Inheritance, Numbers and Strings, Generics (4 hrs)
Inheritance: Extending Classes and Inheritance, Types of Inheritance in Java, Polymorphism, Type Compatibility and Conversion, Overriding and Hiding Methods, Hiding Fields, Using the Keyword “super”
Numbers and Strings: String Class and Methods, StringBuffer Class and Methods
Generics: Generic Classes and Methods.

Unit 3: Exceptions, Error Handling and Basic I/O (5 hrs)
Exceptions and Error Handling: Exceptions and Errors, Catching and Handling Exceptions The try Block, The catch Blocks, The finally Block, Specifying the Exceptions Thrown by a Method, Throwing Exceptions, Chained Exceptions , Creating Exception Classes, Checked and Unchecked Exceptions, Advantages of Exceptions.
Basic I/O: I/O Streams, Byte Streams, Character Streams, Buffered Streams, Scanning and Formatting, Data Streams, Object Streams , File I/O Classes: Reading, Writing, and Creating
Files and Directories.

**Unit 4: Java Collections Framework** *(3 hrs)*

**SECTION II**

**Unit 5: Multithreading and Network Programming** *(4 hrs)*
**Multithreading:** Creating Threads, Thread scheduling and priority, Thread interruptions and synchronization, Thread Safety, Pros and Cons of Multithreading.
**Network Programming:** Networking fundamentals, TCP, UDP communication in Java. Client server programming: InetAddress, URLs, Sockets, DatagramSockets.

**Unit 6: JDBC and RMI** *(4 hrs)*
**JDBC:** Introduction to JDBC API.
**RMI:** Introduction, RMI the Server and Client, Re object, Writing and Skeleton Classes.

**Unit 7: GUI Programming** *(4 hrs)*
**GUI Programming with** package, , Layouts, Event Applets: Introduction, Developing and Deploying Applets

**Unit 8: Servlets and JSP** *(4 hrs)*
Introduction to Servlets and JSP, Servlet architecture and lifecycle. JSP Elements. Request and Response Objects in Servlet API and JSP, Cookies and Session Handling using Servlet.

**Text Books:**
1. Head First Java – Kathy Sierra, Bert Bates, O’Reily Publication
2. The Java™ Programming Language By Ken Arnold, James Gosling, David Holmes, Pearson Publication
3. Head First Servlets and JSP – Bryan Bosham, Kathy Sierra, Bert Bates, O’Reily Publication

**Reference Books:**
Term Work:

- Students should undertake minimum 20 practical assignments based on each above topic.
- The assignments should test and develop student’s practical proficiency and ability to use Java API Classes efficiently in writing effective code for varied applications scenarios & requirements.
- Use of IDEs like BlueJ, Eclipse, Netbeans for Interactive development and debugging of Java applications is highly recommended to enhance hands on skills in Java Programming of Students.
- Preferably use Apache Tomcat/GlassFish Server with Eclipse or Netbeans for assignments based on Servlets and JSP.
Refer to the syllabus common to all programmes under faculty of engineering.
SOLAPUR UNIVERSITY, SOLAPUR
T.E. (Computer Science & Engineering)
Semester-II

1. COMPILER CONSTRUCTION

Teaching Scheme | Examination Scheme
-----------------|---------------------
Lecture: 3 Hrs/Week | Theory - 100 Marks
Practical: 2 Hrs/Week | Term-Work – 25 Marks

COURSE OBJECTIVES:
1. To introduce principal structure of compiler, basic theories and methods used for different parts of compiler.
2. To impart knowledge of fundamentals of language translator, structure of a typical compiler, parsing methods etc.
3. To design various phases of compiler such as Lexical analyzer, parser etc.
4. To distinguish different optimization techniques in the design of compiler.

COURSE OUTCOMES:
At the end of the course, student will be able to
1. Apply techniques for the structure of compiler.
2. Use simulation software to justify compiler design.
3. Implement various phases of compiler.
4. Apply different optimization techniques in the design of compiler.
5. Analyze and compare various compilers to select optimum.

Course instructions:
Study of open source software – YASS by University of Wisconsin-Madison, to better understand each phase of compiler.

SECTION - I

Unit 1: Introduction: (3 Hrs.)
Language Processor, Structure of Compiler

Unit 2: Lexical Analysis: (5 Hrs.)
The Role of the Lexical analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, Finite Automata, Conversion of NFA to DFA, Designing a Lexical Analyzer Generator- The Structure of the Generated Analyzer, Pattern Matching based on NFA’s, DFA for Lexical Analyzer.

Unit 3: Syntax Analysis: (8 Hrs.)
Introduction, Context Free Grammars, Writing a Grammar, Top-down Parsing, Bottom-Up Parsing, Introduction to LR parsing: Simple LR, More powerful LR parsers

Unit 4: Syntax Directed Translation: (6 Hrs.)
Syntax Directed Definitions, Evaluation Order for SDD’s, Application of Syntax Directed Translation, Syntax Directed Translation Scheme, Bottom-up Parsing of L-Attributed SDD’s.
SECTION - II

Unit 5: Intermediate Code Generation: (6 Hrs.)
Variants of Syntax Trees, Three Address Code, Types & Declaration, Control Flow – Boolean Expression, Short Circuit Codes & Flow of Control Statements, Backpatching, Switch-Statements.

Unit 6: Run Time Environments: (3 Hrs.)
Storage Organization, Stack Allocation of Space.

Unit 7: Code Generation: (7 Hrs.)

Unit 8: Code Optimization: (7 Hrs.)

Text Book:
2. Compiler Construction - Dhamdhere (Mc-Millan)

References:
2. Compiler Design in C – Allen I. Holub (PHI / Pearson Education)
3. Compiler Construction - Barret, Bates, Couch (Galgotia)
4. Crafting a compiler with C – Charls Fischer, Richard LeBlane (Pearson Education)

Term Work:
It should consist of minimum 8-10 experiments based on the above topics. Following experiments may be conducted for the term work.

Practical List
1. Design a lexical analyzer for a language whose grammar is known.
2. Implement a recognizer for the language in 1.
3. Recursive Descent Parser
4. Shift Reduce Parser
5. Operator Precedence Parser
6. Generate a symbol table for the language given in 1.
7. Generate 3 address codes for the language given in 1.
8. Implement code optimization techniques on the code produced in 7.
9. Generate target code for the code optimized in 4, considering the target machines to be X86.
10. Code Optimization Tools
SOLAPUR UNIVERSITY, SOLAPUR
T. E. (Computer Science & Engineering)
Semester – II

2. UNIX OPERATING SYSTEM

COURSE OBJECTIVES:
Teacher needs to focus on and make students learn about some of the following things but not limited to:
1. To introduce fundamentals and architecture of UNIX Operating system including file management, process management, memory management and I/O subsystem of UNIX.
2. To provide hands on commands of UNIX and Shell Programming
3. To build the concept of multiuser operating system.

COURSE OUTCOMES:
At the end of the course, student will be able to
1. Illustrate File Structure, Process Management and Memory Management of Unix using UNIX Architecture
2. Apply basic UNIX/Linux commands, system calls and SHELL Programming
3. To compare between single user and multiuser system

SECTION – I

Unit 1: Introduction (6 Hrs.)

Unit 2: The Buffer Cache (6 Hrs.)
Buffer headers, structure of the buffer pool, scenarios for retrieval of a buffer, reading and writing disk blocks, advantages and disadvantages of cache.

Unit 3: Internal Representation of Files (4 Hrs.)
Inodes, structure of the regular file, directories, conversion of a pathname to inode, super block, inode assignment to a new file, allocation of disk blocks, other file types.

Unit 4: System calls for the file System (5 Hrs.)
SECTION – II

Unit 5: The Structure of process (4 Hrs.)
Process stages and transitions, layout of system memory, the context of a process, Saving context of a process, manipulation of the process address space.

Unit 6: Process Control (6 Hrs.)
Process creation, signals, process termination, awaiting process termination, invoking other programs, the user id of a process, the shell, system Boot and the Init process.

Unit 7: Process Scheduling and Time (3 Hrs.)
Process Scheduling, system call for time, clock.

Unit 8: Memory management policies (5 Hrs.)
Swapping, Demand paging

Unit 9: The I/O Subsystem (4 Hrs.)
Driver interfaces, disk drives, terminal drivers, Streams.

Text Books:
1. The design of Unix Operating System - Maurice J. Bach (PHI)
2. Unix Manuals.

Reference books:
3. UNIX Concepts & Applications by Sumitabha Das

Term Work:
It should consist of minimum 8-10 experiments based on the above topics. Following experiments may be

Practical List:
1. Study of Unix Architecture.
2. Write a program for file copy.
3. Write a program for file transfer.
4. Write a program to implement ls command
5. Write a program to implement getblk algorithm
6. Write a program to implement ialloc & ifree algorithm.
7. Write a Program to implement alloc and free algorithm.
8. Study of System calls STAT & FSTAT, PIPES, LINK & UNLINK, DUP, MOUNT & UNMOUNT.
9. Study of shell programming
   • WAP to find whether entered number is even or odd
   • WAP to find factorial of number
   • WAP to find whether entered number is prime or not
   • WAP for fibonacci series
   • WAP to find sum of series of entered number
   • WAP to find power of number.
10. WAP to implement malloc algorithm.
SOLAPUR UNIVERSITY, SOLAPUR
T.E. (Computer Science and Engineering)
Semester-II
3. MOBILE COMPUTING

Teaching Scheme
Lecture: 3 Hrs/Week
Tutorial: 1 Hr/Week

Examination Scheme
Theory: 100 Marks
Term-Work: 25 Marks

COURSE OBJECTIVES:
1. To introduce concepts and principles of mobile computing.
2. To explore skills of finding solutions for mobile computing applications.
3. To get acquainted with basics of Android Operating System and its architecture.
4. To introduce NFC standards and practices.

COURSE OUTCOMES:
At the end of the course, student will be able to
1. Apply the principles of mobile computing in the real time.
2. Analyze requirements of mobile compatible applications.
3. Put the basic knowledge gained, into practice in developing mobile based applications using Android.
4. Analyze various scenarios and environments, where NFC can be put into practice.

Course Instruction
Visit to BSNL for practical working of wired and wireless communication system.

SECTION I

Unit 1: Introduction to Mobile Communication (3 Hrs.)
Analog Communication: Carrier signal, AM, FM, PM, Demodulation, Generations: 1G, 2G, 3G and 4G

Unit 2: Wireless Transmission (6 Hrs.)
 Frequencies for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular system, SDMA, FDMA, TDMA, CDMA.

Unit 3: GSM (7 Hrs.)

Unit 4: Wireless LAN (7 Hrs.)
IEEE 802.11, Personal Area Network, IEEE 802.15.1 and IEEE 802.15.4 (Bluetooth and ZigBee), Ad-hoc and Sensor network-Introduction, Characteristics of MANET and Applications.

SECTION II

Unit 5: Mobile Network Layer (7 Hrs.)
Mobile IP, DHCP

Unit 6: Mobile Transport Layer (7 Hrs.)
Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast and selective retransmission and recovery, Transaction oriented TCP.
Unit 7: Android OS (Case Study)  
Introduction, History, Features and Characteristics, Ecosystem, Hardware Requirements,  

Unit 8: Near Field Communication (Case Study)  
Towards NFC Era, Ubiquitous Computing, Technological Motivation of NFC, RFID and NFC,  
General architecture of NFC enabled mobile phones.

Text Books:
1. Mobile Communications – Jochen Schiller (PEARSON) (Chapters: 2,3,4,5,6)  
2. Introduction to Wireless and Mobile System-D.P.Agrawal and Qing-AnZeng (CENGAGE)  
   (Chapter:1,4)  
3. Embedded Android-Porting, Extending, and Customizing- Karim Yaghmour (O'Reilly Media) (Chapter:7)  
4. Near Field Communication: From Theory to Practice – Vedat Coskun, Kerem Ok, Busra Ozdenizci. (Wiley) 

Reference Books:
1. Wireless Communication – Theodore S. Rappaport (PEARSON)  
2. Mobile and Personal Communication Systems and Services - Raj Pandya – (PHI)  

Termwork:
Assignment List:
1. The message signal $x(t)=\sin(100t)$ modulates the carrier signal $c(t)=A \cos(2\pi fct)$. Using  
   amplitude modulation, find the frequency content of the modulated signal. 
2. Compare and discuss the various techniques used in Multiple Division Techniques. 
3. A TDMA system uses a 270.833Kbps data rate to support eight users per frame.  
   a) What is the raw data provided for each user?  
   b) If guard time and synchronization occupy 10.1Kbps, determine the traffic efficiency. 
4. Give reasons for a handover in GSM and the problems associated with it. What are the  
   typical steps for handover, What types of handover can occur? 
5. Which resources need to be allocated during handover for data transmission using HSCSD  
   or GPRS respectively? What about QoS guarantees? 
6. How do IEEE 802.11, HiperLAN2 and Bluetooth, respectively, solve the hidden terminal  
   problems? 
7. List the entities of mobile IP and describe data transfer from a mobile node to a fixed node  
   and vice versa. Why and where is encapsulation needed? 
8. What is the basic purpose of DHCP? Name the entities of DHCP. How can DHCP be used  
   for mobility and support of mobile IP? 
9. How and why does I-TCP (Indirect TCP) isolate problems on the wireless link? What are  
   the main drawbacks of this solution? 
10. Write a case study for selected Android OS of specific version. 
11. Case study of NFC in modern smart phone mobiles.
SOLAPUR UNIVERSITY, SOLAPUR
T.E. (Computer Science & Engineering)
Semester-II

4. DATABASE ENGINEERING

<table>
<thead>
<tr>
<th>Teaching Scheme</th>
<th>Examination Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture: 4 Hrs/Week</td>
<td>Theory: 100 Marks</td>
</tr>
<tr>
<td>Practical: 2 Hrs/Week</td>
<td>Termwork: 25 Marks</td>
</tr>
<tr>
<td></td>
<td>Practical Oral Exam. 50 Marks</td>
</tr>
</tbody>
</table>

---

COURSE OBJECTIVES:
1. To develop the relational model of data,
2. To introduce the students an overview of the database-design process, with E-R model and develop query writing skills in SQL.
3. To familiarize the students with concept of normalization of database.
4. To express the fundamentals of a transaction-processing system and concurrency control.

---

COURSE OUTCOMES:
At the end of this course, the student will be able to,
1. Apply the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and database language SQL.
2. Design E-R diagrams to represent simple database for any real time application and formulate SQL queries on it.
3. Design a database, analyze it and improve the design by normalization.
4. Demonstrate knowledge of ACID properties of a transaction and several techniques of concurrency control.

---

SECTION-I

Unit 1: Introduction (4 Hrs)

Unit 2: Relational Model (10 Hrs)
Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus.

Unit 3: Database Design and the E-R Model (6 Hrs)
Unit 4: Relational Database design (8 Hrs)

SECTION - II
Unit 5: Indexing and Hashing (7 Hrs)
Basic Concepts, Ordered Indices, B⁺-Tree Index Files, B⁺-Tree Extensions, B Tree Index Files, Multiple Key Access, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Bitmap Indices, Index Definition in SQL.

Unit 6: Transactions (7 Hrs)
Transaction Concept, A Simple Transaction Model, Transaction Atomicity and Durability, Transaction Isolation Levels, Transaction Isolation as SQL Statements.

Unit 7: Concurrency Control (6 Hrs)
Lock-Based Protocols, Deadlock Handling, Multiple Granularity, Timestamp-Based Protocols, Validation-Based Protocols.

Unit 8: Recovery System (8 Hrs)
Failure Classification, Storage Management, Log-Based Transactions, Buffer Management, Log-Based Recovery, Shadow Paging, Recovery with Concurrent Transactions, Failure with Loss of Nonvolatile Storage.

Text Books:

Reference books:
3. Principles of Database Systems by J. D. Ullman (Galgotia Publications)

Course Instructions:
Assignments 1 to 6 should be implemented in PostGreSQL/MySQL/Oracle.
Assignments 7 to 11 should be implemented in C++/Java.
**Term Work:**  
It should consist of 8-10 laboratory assignments as follows:

1. E-R Diagrams (around 5 in number) for any specific application and create a data dictionary for the same.
2. Basic SQL-write simple queries in SQL on the schema created for a specific application.
3. a) More SQL: Aggregates-write queries in SQL using aggregates, grouping and ordering.  
   b) Nested sub queries and SQL updates: write queries in SQL using concept of nested sub queries and SQL update commands.
4. a) SQL DDL
   b) Schema create const
5. Convert the
6. Write a Java
   using JDBC
7. Write a pro
   on the database
8. Write a pro
   database previously
9. Write a pro
   iate or deferred
10. Write a pro
11. Given a se
    functional d
5. SOFTWARE ENGINEERING

COURSE OBJECTIVES
The Course should enable the student
1. To focus on the study of plan, design, architecture and modeling structure layout of software.
2. To illustrate and compare use of life cycle models of software development.
3. To enable the students to analyze and estimate the cost, effort of software product.
4. To learn to embed various quality standards in the software.

COURSE OUTCOMES
At the end of the course, student will be able to
1. Develop the software project using appropriate phases.
2. To implement life cycle models in software development and for their projects.
3. To enhance the quality of product and should be able to apply testing of software.
4. Know the basics of software metrics and result assessment and basics of process improvement.

SECTION-I

Unit 1: Introduction to Software Engineering: (6 Hrs.)

Unit 2: Software Requirements Analysis and Specification: (4 Hrs.)
Software Requirements, Problem Analysis, Requirements Specification, Validation, Metrics.

Unit 3: Function and Object Oriented Design (8 Hrs.)
Design Principles, Module Level Concepts, Design Notation and Specification, Structured Design Methodology, Verification, Metrics.
OO Analysis and OO design Concepts, Design Notation and Specification, Design Methodology, UML Diagrams.

Unit 4: The Project Planning (5 Hrs.)
SECTION-II

Unit 5: Quality planning, Risk Management and Tracking
(5 Hrs.)

Unit 6: Agile Project Management
(5 Hrs.)

Unit 7: Managing Software projects, Project execution and closure
(8 Hrs.)

Unit 8: Testing
(5 Hrs.)
Testing Fundamentals

Text Books:
2. Effective Project Management: Traditional, Agile, Extreme, Robert K. Wysocki WILEY INDIA, 6th edition
3. Software Project Management in Practice: Pankaj Jalote

Reference Books:
1) Ian Sommerville. Software Engineering
2) Software Engineering Fundamentals – Ali Behforooz and Frederick J. Hudson
3) PAKAJ JALOTE’S Software Engineering, A Precise Approach (Wiley Precise Textbook, WILEY INDIA)
4) Software Engineering by Ian Sommerville.

Termwork:
Tutorial List:
Implémentation of mini software projects by applying SDLC cycles.
It should consist of minimum 6 - 8 assignments based on each topic of above syllabus.
COURSE OBJECTIVES:
1. To introduce .NET Programming using the C# programming language.
2. To develop basic understanding of the syntactical features of C# programming language and effective use of .NET runtime library APIs to develop robust software applications.
3. To develop ability to design and build Object Oriented and GUI, Web applications on Windows platform.

COURSE OUTCOMES:
At the end of the course, student will be able to
1. Use .NET Framework in building robust software applications using C# programming language.
2. Design and develop Object Oriented and GUI, Web application on Windows platform.

SECTION I

Unit 1: Introduction to .NET Framework (3 Hrs)
The .NET architecture, The common language runtime (CLR), the, Microsoft intermediate Language code (MSIL), Just in time Compliers, The framework class library, The common language specification, common language type system (CTS), Introduction to Visual Studio .NET and Sharp Develop IDE.

Unit 2: C# Application Basics and Language fundamentals (4 Hrs)
Creating and compiling C# programs using command line compiler (csc.exe), Creating applications using IDEs, Namespaces, the “using” keyword, Basic data types, Operators, Flow control and conditional statements, loops, Arrays, Classes and Objects, Constructor overloading, Methods, Fields, Properties, Access Modifiers and Accessibility Levels, Static methods and fields, Garbage Collection, Structures, Nested Classes, String Manipulations, Naming Conventions, Java vs. C#.

Unit 3: Object Oriented Programming using C# (4 Hrs)
Objects and Reference Types, Inheritance, Interfaces and Abstract Classes, Polymorphism, the “virtual” and “override” keyword, the “base” keyword, the “sealed” keyword, The Object Class, the “new” keyword in context of method overriding, Type Casting: Up casting and Down casting, the “is” and “as” keywords, Boxing and Unboxing.

Unit 4: Exception Handling, Events and Delegates (4 Hrs)
Need for Exceptions, Exception Hierarchy, Handling Exceptions using try-catch-finally blocks, creating and defining Custom Exceptions, the “throw” keyword. Events and Delegates in C#, Multicast Delegate, Event Handling.
SECTION II

Unit 5: Multithreading and Basic IO in C#  
(4 hrs)
What is Multithreading, Multithreading in C#, Static and Instances members of Thread Class, Basic Thread operations, Thread priorities, Thread Synchronization, 
**File System and Streams:** Streams and System.IO namespace, Console IO, Reading writing and updating files and directories, System.IO.FileInfo Class, Serialization and Deserialization.

Unit 6: GUI Programming in C#  
(4 hrs)

UNIT 7: Data access  
(4 hrs)
Introduction to ADO.NET, System Data namespace, Data Table, Data Row, Data Column and other prominent classes, Accessing and Updating Data using ADO.NET.

UNIT 8: Introduction ASP.NET  
(4 hrs)
Introduction to ASP.NET, State management in ASP.NET, ASP.NET Web Forms, Server Controls, Web application configuration, Creating Web applications using ASP.NET and C#.

Textbooks:

Reference Books:

Term Work:
- Students should undertake minimum 10 practical assignments based on each above topic.
- The assignments should test and develop student’s practical proficiency and ability to use .NET framework libraries and APIs efficiently in writing C# code for varied applications scenarios & requirements.
- Use of IDEs like SharpDevelop and Visual Studio Express Edition for Interactive development and debugging of C#.NET applications is highly recommended to enhance hands on skills in C#.NET Programming of Students.
7. Mini Project

COURSE OBJECTIVES:
1. To undertake investigation of complex problems.
2. To motivate students to undertake design of a product, which is sustainable and meaningful to society.
3. To enable students to acquire and develop professional skills.
4. To make students learn to work in team.
5. To encourage independent critical thinking, creativity and discipline.
6. To use modern tools and simulation packages.
7. To prepare students to implement their acquired engineering knowledge for society.

COURSE OUTCOMES:
At the end of the course, student will be able to
1. Identify and define the problem.
2. Develop a sustainable product or offer a effective solution to industrial problem.
3. Present proposal within budgetary and time constraints with effective communication and writing skills.
4. Develop leadership qualities.
5. Criticize and refine own solution or product.
6. Apply modern tools and simulation packages to develop product.
7. Develop a strong sense of social responsibility and accountability.

Note:
1. There should be a group of preferably 4 students.
2. Students should be given projects in hardware, software, embedded or any contemporary topic in CSE and/or IT.
3. One guide should be allocated per batch.

Mini Project ideas (but not limited to):
1. Online Examination module (Multiple choice questions)
2. Attendance recording and analysis software module
3. Examination Result analysis software module
4. Hardware exhibitors such as display board exhibiting all types of mouse / keyboards, HDDs, Monitors etc.), Internal architecture and working
5. Departments / College website
6. Library Management System
7. Hotel Management System
8. Time table generation
9. CD Library management system
10. Admission procedure automation
11. Online passport registration automation
12. Student Feedback system automation
13. Ice Cream parlor management system
14. Pizza hut – account management system
15. Multi player strategy game – Project ideas on Visual Basic, Java, Database
16. A speech response application using some hardware interface using the Microsoft SAPI SDK
17. LAN administrator tool (socket programming comes easy in VB) which will monitor application on a LAN and provide functions.
18. Voice mail systems
19. Computer telephony integration
20. Student Information System
21. Traffic Control system
22. Airline reservation system
23. Simulation for Balloon shooting game
24. Mini Calculator in 'C'
25. Moving ball game
26. Tic-tac-toe game
27. Design a personal profile web page
COURSE OUTCOMES:
At the end of the course, student will be able to
1. Learn Modeling and Simulation.
2. Use API libraries for Network Simulator.
3. Perform a task completely on Network Simulator.

Unit 1: Introduction to Modeling and Simulation:
When simulation is the appropriate tool and when it is not appropriate; Advantages and disadvantages of Simulation; Areas of application; Systems and system environment; Components of a system; Discrete and continuous systems; Why and what to Model, Model of a system, Types of Models,

Unit 2: Case study of NS-x
Purpose of NS2, Overview, OTcl: The User Language, Simple Simulation Example, Event Scheduler, Network Components, Packet, Post Simulation: Trace Analysis and Examples, Types of Queue Monitor and Examples,

Unit 3: Basic Scenarios using NS2
Writing OTcl code for following:
1. Scenario for different topologies – star, bus, mesh, ring
2. Wired Scenario for different bandwidth and packet size for 10 nodes for LAN.
3. Scenario for TCP and UDP with proper example.
4. Comparative Graph for any two scenarios.

Books and References:
2. Simulation Modeling and Analysis – Averill M. Law
3. http://nile.wpi.edu/NS/ -- NS by Example tutorial
4. Network Simulator website and NS2 manual
COURSE OUTCOMES:
At the end of the course, student will be able to
1. Compare between various components of network, select appropriate network topology and setup a computer network.
2. Use the network management tools for monitoring the network performance.
3. Demonstrate different ways of network setup and use of network management tools.

Unit 1: Computer Network Setup

Unit 2: Network Management
Network Management Architectures and Applications, Configuration Management and Auto Discovery, Configuration Databases and Reports, Abstract Syntax Notation One (ASN.1)

Unit 3: Network Management Functions

Unit 4: Management Tools, Systems and Applications

Books and References:
1. Networking – The Complete Reference by Craig Zacker Tata McGraw Hill (Unit 1)
COURSE OUTCOMES:
At the end of the course, student will be able to
1. Present software licensing models and practices adopted in software development and distribution.
2. Analyze, compare and choose appropriate software licensing model and strategy for the software developed.

Unit 1: Introduction

Unit 2: Software Licenses
The MIT License, The BSD License, The Apache License, v1.1 and v2.0, The Academic Free License, Application and Philosophy of MIT and BSD Licenses, GNU General Public License, GNU Lesser General Public License, The Mozilla Public License, Application and Philosophy of GNU GPL and GNU LGPL.

Unit 3: Creative Commons Licenses and Non Open Source Software Licenses
Creative Commons Licenses, Classic Proprietary License, Sun Community Source License, Microsoft Shared Source Initiative.

Unit 4: Legal Impacts of Open Source and Free Software Licensing

Textbooks:

Reference Books: