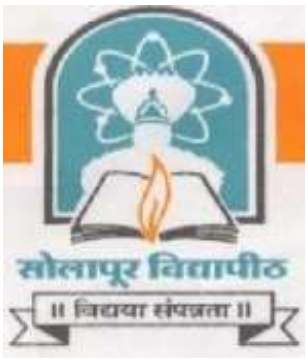

SOLAPUR UNIVERSITY, SOLAPUR

FACULTY OF ENGINEERING & TECHNOLOGY

COMPUTER SCIENCE & ENGINEERING

Structure & syllabus for

T.E. (Computer Science & Engineering)
w.e.f. Academic Year 2018-19

Choice Based Credit System

Solapur University, Solapur

Faculty of Engineering and Technology

Program : Computer Science and Engineering

A) Program Educational Objectives

1. To provide students good knowledge of Mathematics, Science and Technology as well as the logical base of Computer Science that will be useful in solving complex engineering problems and develop lifelong learning ability.
2. To impart knowledge with good understanding of fundamentals of all subjects of Computer Science and Engineering, so that students are able to analyse, design and implement new projects from various application domains using various modern engineering tools.
3. To develop excellent logical thinking & programming skills to enable students to design, develop system and application level softwares within realistic constraints.
4. To make students good human beings who will have sense of social responsibility and respect over society & its heritage by creating good social environment for them as well as teach them professional and ethical standards.
5. To improve communication, presentation, team working skills and managerial skills leading to entrepreneurship and leadership.
6. To introduce students with new technology to meet the challenges of changing scenario in IT Sector and make them aware of contemporary issues at national and international level.

B) Programme Outcomes (POs)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

C) Program Specific Outcomes (PSOs)

1. Graduate has an ability to use technical skills necessary for design, maintenance, development and implementation of database systems and networking applications.

2. Graduate has an ability to provide IT Solutions develop mobile applications in multidisciplinary areas using standard tools and techniques.

3. Graduate has an ability to utilize and apply software engineering tools for design and realization of projects in various domains of Computer Science & Engineering.

Choice Based Credit System Syllabus Structure of T.E.Computer Science and Engineering W.E.F. 2018-2019 ***Semester I***

Course Code	Theory Course / Name	Hrs./Week			Credits	Examination Scheme			
		L	T	P		ISE	ESE	ICA	Total
CS311	Operating System Concepts	\3	--	---	3	30	70	--	100
CS312	System Programming	3	--	---	3	30	70	--	100
CS313	Database Engineering	3	--	---	3	30	70	-	100
CS314	Design and Analysis of Algorithms	3	1	---	4	30	70	25	125
CS315	Computer Organization	3	1	---	4	30	70	25	125
CS316	Java Programming	3	---	---	3	--	--	--	--
SLH31	Self Learning Module 1	--	---	---	2	--	50	--	50
	Sub Total	**18**	**02**		**22**	**150**	**400**	**50**	**600**
	Laboratory								
							ESE		
							POE	OE	
CS311	Operating System Concepts	---		2	1	---	50	--	25	75
CS312	System Programming	---		2	1	---	---	--	25	25
CS313	Database Engineering	---		2	1	---	50	--	25	75
CS316	Java Programming	---		4	2	---	50	--	25	75
	Sub Total	**18**	**02**	**10**	**5**	**150**	**150**		**100**	**250**
	Grand Total				**27**		**550**		**150**	**850**

- *Abbreviations: L- Lectures, P –Practical, T- Tutorial, ISE- In Semester Exam, ESE - End Semester Exam,ICA-Internal Continuous Assessment, ESE - University Examination (Theory &/ POE &/Oral examination)*

Choice Based Credit System Syllabus Structure of T.E.Computer Science and Engineering W.E.F. 2018-2019 ***Semester II***

Course Code	Theory Course / Name	Hrs./Week			Credits	Examination Scheme			
		L	T	P		ISE	ESE	ICA	Total
CS321	Compiler Construction	4	--	--	4	30	70	--	100
CS322	Unix Operating System	3	--	--	3	30	70	--	100
CS323	Mobile Computing	3	1	--	4	30	70	25	125
CS324	Software Engineering	3	1	--	4	30	70	25	125
CS325	Mobile Application Development	3	--	_	3	30	70	--	100
CS 326	Programming in C# net	2	--	--	2		---	--	--
SLH 32	Self Learning Module 2	--	--	--	2		50	--	50
	Sub Total	**18**	**02**		**22**	**150**	**400**	**50**	**600**
	Laboratory								
							ESE		
							POE	**OE**	
CS321	Compiler Construction	--	--	2	1	---	--	---	25	25
CS322	Unix Operating System	--	--	2	1	---	---	---	25	25
CS325	Mobile Application Development	--	--	2	1	---	50	---	25	75
CS326	Programming in C# net	--	--	2	2	---	50	---	25	75
CS327	Mini Project	--	--	2		---	50	---	---	50
	Sub Total	**18**	**--**	**10**	**5**	**---**	**150**		**100**	**250**
	Grand Total	**18**	**02**	**10**	**27**	**150**	**550**		**150**	**850**

- *Abbreviations: L- Lectures, P –Practical, T- Tutorial, ISE- In Semester Exam, ESE - End Semester Exam,ICA-Internal Continuous Assessment, ESE - University Examination (Theory &/ POE &/Oral examination)*

Self Learning Module 1	**Self Learning Module 2**
Subjects 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 4. Ethical Hacking 5. Data Science 6. UI Technologies

Note:

1. The Internal Continuous Assessment (ICA) 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
4. Student shall select one Self Learning Course at T.E. Part I and T.E. Part II each from 'Humanities & Social Sciences (HSS) ' and 'Technical' Group respectively.
5. For TE Part I -
   A. Student can select a Self Learning Course from Solapur University, Solapur HSS Course List and appear for its examination as and when conducted by Solapur University, Solapur.

   OR

   B. Student can enroll for National Programme on Technology Enhanced Learning (NPTEL) course, complete its assignments and appear for certificate examination as and when conducted by NPTEL.

   For more details about Self Learning Course (HSS) please refer to separate rule document available from Solapur University, Solapur
   More details about NPTEL are available at http://nptel.ac.in
6. Project group for T.E.(CSE) Part II Mini Project shall be of 4 / 5 students
7. 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
8. Curriculum for Humanities and Social Sciences Self Learning Modules is common for all under graduate programmes of faculty of Engineering and Technology.

Solapur University, Solapur
T.E. (Computer Science & Engineering)
Semester I
CS311 – OPERTING SYSTEM CONCEPTS

Teaching Scheme	**Examination Scheme**
Lectures– 3 Hours/week, 3 Credits	ESE – 70 Marks
Practical – 2 Hour/week, 1 Credits	ISE – 30 Marks
	ICA - 25 Marks
	POE – 50 Marks

Introduction:
This course introduces Fundamentals and basic knowledge of an operating system. It also covers the details Process Management, deadlock, Memory Management and IO subsystems.

Course Prerequisite: Students should have knowledge of Computer Systems and basics of C programming language.

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:
Students 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)**
Operating system definition, Simple Batch System, Multiprogrammed Batch System, Time Sharing System, Personal Computer System, Parallel System, Real Time System, and System Calls.

Unit-2 Process **(6 hrs)**
Process Concept, Process Scheduling, Operations on processes, Cooperating Processes, Threads, Inter-Process communication

Unit-3 Process Scheduling **(6 hrs)**
Basic concept, Scheduling Criteria, Scheduling Algorithms, Multiple processor scheduling, Real time scheduling (Algorithms evaluation).

Unit-4 Inter-process synchronization **(5 hrs)**
Background, The critical section problem, Peterson's algorithm, 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 IO System **(4 hrs)**

Overview, I/O hardware, Application I/O interface, Kernel I/O subsystem, Transforming I/O request to hardware operation.

ICA

It should consist of 10 to 12 experiments based on C and Linux.

1. Study of UNIX operating system and its commands.
2. Implementation of a program which describe the use of system calls such as fork (), abort (), suspend () etc.
3. Implementation of FCFS scheduling algorithm.
4. Implementation of SJF (preemptive & non preemptive)
5. Implementation of round robin (RR).
6. Implementation of priority scheduling algorithm.
7. Implementation of Banker's Algorithm for Deadlock Avoidance.
8. Implementation of RAG or WFG method for Deadlock detection for single instance of resources.
9. Simulation of Page Replacement strategies (FIFO, LRU, Optimal) based on Java Multithreading.
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).

Text Books:

1. Operating System concepts – 7th or 8th Edition – Silberschatz, Galvin (John Wiley).

Reference Books:

1. Operating Systems: Internals and Design Principles, 5th Edition by William Stallings (PHI).
2. Operating system with case studies in UNIX, Netware and Windows NT by Achyut Godbole (TMGH).
3. Operating Systems – Deitel, Deitel, Choffnes – 3rd Edition, by Pearson Education.

Solapur University, Solapur
T.E. (Computer Science & Engineering)
Semester I
CS312 – SYSTEM PROGRAMMING

Teaching Scheme	**Examination Scheme**
Lectures – 3 Hours/week, 3 Credits	ESE – 70 Marks
Practical – 2 Hour/week, 1 Credits	ISE – 30 Marks
	ICA - 25 Marks

Introduction:
This course introduces Language Processing activities, which helps to understand the basics of design and development of various profession languages, along with understanding of all the system software involved in executing a particular code written in a particular language.

Course Prerequisite: Students should have knowledge of Data Structures, Computer Organization, Microprocessors, Advanced 'C' Concepts.

COURSE OBJECTIVES:

1 To learn and understand fundamentals of System Software Programs as Assembler, Macro-processor, Linkers and Loaders.
2 To design and develop various System Software Programs.
3 To acquire skills of Language processor development tools.

COURSE OUTCOMES:
Students will be able to**:**

1. Identify the requirement of different System Software for the execution of application software.
2. Design and implement various System Programs Assembler and Macros.
3. Recognize the importance of language processing development tools in formal language implementation.
4. Demonstrate the working principles of System Software.

SECTION-I

Unit-1 Language Processors **(8 hrs)**
Introduction, language processing activities, Fundamentals of language processing, Fundamentals of language, Specification, language Processor development tools,UNIX programming Tools-lex & yacc, Recognizing words with Lex, Parser lexer communication, the parts of Speech lexer, A Yacc parser, The rules section of yacc, running lex and yacc on Unix.

Unit-2 Assemblers **(7 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 **(7 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.

ICA:

Student should perform minimum 8 experiments based on the following guidelines.

1. Design Lex specifications for the tokens – keywords, identifiers, numbers, operators, white spaces.
2. Implementation of simple Lexical Analyzer in C which will generate the different tokens.
3. Implementation of syntax recognizer using grammar rules.
4. Simulation of text editor.
5. Introduction of TASM.
6. Symbol Table generation for *.c or *.asm file.
7. Design and Implementation of two pass assembler.
8. Design and Implementation of Single pass assembler.
9. Implementation of Macros.
10. Implementation of Nested macros.
11. Implementation of Toy-code generator.
12. Simulation of linkers.
13. Simulation of loaders.

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)
3. Unix Programming Tools – lex & yacc , John R. Levine, Tony Mason & Doug Brown, (O'REILLY) (Unit 1)

Reference Books:

1. System Software- An Introduction to Systems Programming- 3rd Edition- Leland L. Beck (Pearson Education)
2. Adam Hoover,"System Programming with C and Unix", Pearson,2010
3. Terence Parr, "Language Implementation Patterns",SPD,2009

Solapur University, Solapur
T.E. (Computer Science & Engineering)
Semester I
CS313 – DATABASE ENGINEERING

Teaching Scheme
Lectures: 3 Hrs/week, 3 Credits
Practical: 2Hrs/week, 1 Credit

Examination Scheme
ESE: 70 Marks
ISE: 30 Marks
ICA: 25 Marks
POE: 50 Marks

Introduction:
In today's data-driven economy, no computer science or business curriculum would be complete without a course in databases and data management system. This course emphasizes the understanding of the fundamentals of relational database system including data models, database architectures, normalization, data integrity, security and data manipulation.
It also provides students with theoretical knowledge and practical skills in the use of databases and database management systems with the help of Structured Query language (SQL).It ends with covering database transaction and recovery concepts. Upon completion, students should be able to design and implement normalized database structures by creating simple database.

Course Prerequisite:
No prerequisite knowledge of databases is required but basic understanding of data-structures and algorithms is assumed. Any general purpose programming language knowledge is needed.

Course Objectives:
1. To understand the basics of database design, structure, implementation and applications.
2. To develop the logical design of the database using data modeling concepts such as entity-relationship diagrams.
3. To understand and use Structured Query Language to query, update, and manage a database.
4. To apply normalization techniques to normalize the database.
5. To familiarize the students with the fundamentals of database transaction processing and learn techniques for concurrency control and recovery methods.

Course Outcomes:
At the end of this course, the student will be able to,
1. Define and apply the basic concepts of database system, design, relational model and schemas.
2. Design principles for logical design of databases, including the E-R method and normalization approach for any real time application.
3. Evaluate, using relational algebra and SQL, solutions to a broad range of query problems in a relational DBMS.
4. Demonstrate an understanding of normalization theory and apply such knowledge to normalize a database.
5. Be familiar with the basic issues of transaction processing (ACID properties), different methods of concurrency control and recovery techniques.

SECTION – I

Unit 1: Introduction **(3 Hrs)**
Database System Applications, Purpose of Database Systems, View of data, Database Languages, Data Storage and Querying, Database Architectures, Database users and administrators, history of databases system.

Unit 2: Database Design and E-R Model (8 Hrs)

Overview of design process, E-R Model, Constraints, Removing redundant attributes in entity sets, E-R diagrams, Reduction to relational schema , E-R design issues, Extended E-R features.
Relational Model : Structure of relational databases, Database schema , keys, Schema diagrams, Relational Query languages, Relational algebra, Tuple Relational Calculus, Domain Relational Calculus.

Unit 3: SQL and Advanced SQL (7 Hrs)

SQL : Overview, SQL data definition, Basic structure of SQL Queries, additional basic operations, Set operations, NULL values, Aggregate functions, Nested sub queries, Modification of the databases , Join operations, Views, Transactions, Integrity constraints, SQL data types and schemas, Authorization, Advanced SQL : Embedded SQL, Functions and Procedures, Triggers.

Unit 4: Relational Database Design (6 Hrs)

Features of good Relational Designs, Atomic Domains, First Normal Form, Decomposition using Functional dependencies, Second Normal Form, BCNF, Third Normal Form, Functional-dependency theory, Fourth Normal Form.

SECTION – II

Unit 5: Indexing and Hashing (4 Hrs)

Basic Concepts, Ordered Indices, B+ Tree Index Files, B Tree Index Files, Multiple Key Access, Static Hashing, Dynamic Hashing, Comparison of Indexing and Hashing, Index definition in SQL

Unit 6: Transactions (5 Hrs)

Transaction concepts, a simple transaction Model, Storage structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity.

Unit 7: Concurrency Control (4 Hrs)

Lock based protocol, Deadlock handling, multiple granularity, Time Stamp-based protocols, Validation based protocols.

Unit 08: Recovery System (5 Hrs)

Failure Classification, Storage, Recovery and Atomicity, Recovery algorithms, Buffer management.

Text Book:

1. Database system concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan (McGraw Hill International Edition) sixth edition.

Reference books:

1. Fundamentals of Database systems by Ramez ElMasri, S. B. Navathe (Pearson Education) Fifth edition.
2. Database Management Systems by Ramkrishnan Gehreke (Tata McGraw Hill) third edition.
3. Principles of Database Systems by J. D. Ullman (Galgotia Publications)
4. SQL The Complete Reference, 3rd Edition by James R Groff, Paul N. Weinberg and Andy Oppel
5. Database system concepts by Peter Rob, Carlos Coronel (Cengage Learning) ninth edition.

Course Instructions:

Assignments 2 to 4 should be implemented in MySQL/Oracle/ PostGreSQL.
Assignments 6 to 11 should be implemented in C++/Java.

Internal Continuous Assessment (ICA) :

It should consist of 8-10 laboratory assignments as follows:

1. E-R Diagrams : Draw E-R diagram for any specific database application and create a data dictionary for the same.
2. a) Basic SQL DDL commands: write simple queries in SQL on above database application for schema creation and updation.

   b) SQL DML commands: insert, update, select command with different clauses, queries using aggregates, grouping and ordering.
3. a) Nested sub queries , Joins and Set operations : write queries in SQL using concept of nested sub queries , join and different set operations.
4. a) Views, Integrity constraints and Authorization : queries for creating views, different integrity constraints and authorization commands.

   b) Advanced SQL: queries on embedded SQL, functions and procedures, triggers
5. Convert the created database into 1NF, 2NF, 3NF and BCNF.
6. Given a set of functional dependencies, find canonical cover and closure of functional dependency.
7. Write a Java program for database (created in expt-2) connectivity using JDBC.
8. Write a program to implement B+ tree index (n=3 or n=5) on the database previously created.
9. Write a program to implement dynamic hashing on the database previously created.
10. Write a program to simulate log based protocol using immediate or deferred database modification.
11. Write a program to simulate any one concurrency control protocol.

Solapur University, Solapur
T.E. (Computer Science and Engineering)
Semester I
CS314 - DESIGN AND ANALYSIS OF ALGORITHMS

Teaching Scheme	**Examination Scheme**
Lectures– 3 Hours/week, 3 Credits	**ESE – 70 Marks**
Tutorial – 1 Hour/week, 1 Credit	**ISE – 30 Marks**
	ISA – 25 Marks

Introduction :
This course introduces the algorithms, strategies of algorithm and analysis of algorithm which will help to compare and determine good algorithm.

Course Prerequisite:
Student should have knowledge of basic programming. They should also have basic knowledge of data structure and graph theory.

COURSE OBJECTIVES:

1. Analyze the Asymptotic Performance of Algorithm (Best,Worst.Average Case).
2. Calculate the time and space complexity of an algorithm.
3. Demonstrate the familiarity with the major Algorithm (Searching and Sorting) .
4. Apply important algorithmic design paradigms and methods of analysis.
5. Apply algorithm design paradigm to solve real life problem.
6. Identify P,NP, NP-complete and NP-Hard Problem and differentiate between tractable and intractable problems.

COURSE OUTCOMES:

1. Student can apply asymptotic rules to simple program/Algorithm.
2. Student can derive time and space complexity of simple to medium level algorithm.
3. Student can be compare and analysis different sorting and searching algorithm
4. Student can able to demonstrate pros and crons of different design paradigms.
5. Student is able to apply known paradigm(s) to solve real life problem.
6. Student is able to differentiate the P, NP, NP-complete algorithm.

SECTION-I

Unit 1 : Introduction (8 Hrs.)
Algorithm Specification: Pseudocode Convntions, Recursive Algorithm, Performance Analysis: Space C mplexity, Time Complexity, Calculating worst case, best case and average case complexities,complexities Asymptotic Notations, Performance Measurement

Unit 2 : Divide and Conquer (7 Hrs.)
The general method, Binary search, Finding the maximum and Minimum, Quicksort, Selection Sort, Merge Sort.

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

Unit 4 : Dynamic Programming (8 Hrs.)
The general method, Multistage graphs, All pair shortest paths, Optimal binary search trees, 0/1 Knap sack, 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- class, NP-complete and NP-hard, Standard NP-complete problems, NP-Hard Problem (Only Basics problems).

Internal Continuous Assessment (ICA) :

Term work consists of minimum eight to ten tutorials based upon above curriculum. Tutorial shall include algorithm and its analysis.

Text Book:

1. Fundamentals of Computer Algorithms–Horowitz, Sahni & Rajasekaran (Galgotia Publications)
2. Fundamental of Algorithm. – Gilles Brassard, Paul Bratley (Pearson Publication)

References:

1. Introduction to Algorithms – Thomas Cormen (Pearson Publication)
2. Introduction to Design and Analysis of Algorithm – By Goodman (McGrawhill)
3. Design and analysis of algorithms - Aho, Hopfcraft and Ullman (Addison wesley)

Solapur University,Solapur
T.E.(Computer Science and Engineering)
Semester I
CS315 - COMPUTER ORGANIZATION

Teaching Scheme
Lecturer- 3 Hr/Week, 3 Credits
Tutorial-1 Hr/Week, 1 Credit

Examination Scheme
ESE -70 Marks
ISE - 30 Marks
ICA- 25 Marks

Introduction :
This course introduces functional Units of digital computer that are responsible for storing and processing information. It also deals with other Units used to bring information into a computer and to send computed results to the outside world.

Course Prerequisite
Students shall have the knowledge of digital logic and number system.

COURSE OBJECTIVES :
1. To expose students to basic concepts of computer organization.
2. To make the students aware of overall design and architecture of computer and its organization.
3. To provide a comprehensive and self contained view of Computer design from hardware point of view.
4. To provide pre-requisites knowledge of advanced computer architecture.

COURSE OUTCOME :
1. Apply instruction sets, addressing modes, and instruction formats for designing and implementing computer based system.
2. Apply mathematical foundations, algorithmic principles and computer science theory in the modeling and design of computer-based systems.
3. Analyze different memory replacement policies for effective utilization of memory.
4. To identify various types of buses, interrupts and I/O operations in a computer system.

SECTION-I

Unit-1 : Introduction to Computer Organization (5 Hr.)
Basic Concepts: Organization and Architecture, Structure and Function, A brief history of Computers.
A top level view: Computer Components, Computer Function-Instruction Fetch and Execute, Interconnection Structures and Bus Interconnection.

Unit-2 : Computer Arithmetic (8 Hr.)
Arithmetic and Logic Unit, Integer Representation, Integer Arithmetic (Negation, Addition, Subtraction, Multiplication and Division),Floating Point Representation(Principles, IEEE Standard for Binary Floating Point Representation,) ,Floating Point Arithmetic (Addition, Subtraction, Multiplication and Division)Instruction Sets: Machine Instruction Characteristics, Types of operands, Intel x86 and ARM data types, Types of Operations, Addressing Modes, Instruction format, RISC,CISC ,RISC Pipelining.

Unit-3 :Computer Memory System (8 Hr.)
Computer Memory System Overview, **Cache Memory:** Cache Memory Principles, Elements of Cache Design: Cache Addresses, Cache size, Mapping function (Direct, Associative and set associative), Replacement Policies (FIFO, LRU, OPT and Stack), Write Policy, Line Size and Number of Caches. **Internal Memory:** Semiconductor Main Memory (Organization, DRAM and SRAM, Chip Logic, Chip Packaging, Module Organization and Interleaved Memory), Flash Memory, Newer Nonvolatile Solid-State Memory Technologies. **External Memory:** Magnetic disk, Solid State Drives, Optical Memory and Magnetic Tape.

SECTION-II

Unit-4 : Input/Output Organization (7 Hr)

External Devices, I/O Module, Programmed I/O , Interrupt Driven I/O, Intel 82C59A Interrupt Controller, The Intel 8255A Programmable Peripheral Interface, Direct Memory Access, Intel 8237A DMA Controller, I/O Channels and Processors, External Interconnection Standards.

Unit-5 : Control Design (6 Hr)

Hardwired Control: Design Methods, State tables, Classical method, One-hot method, Multiplier Control Unit: Microprogram structure Control field encoding, Encoding by function.

Unit-6 : Pipeline and Multiprocessor (7 Hr)

Linear Pipeline Processor: Asynchronous and Synchronous Models, Clocking and Timing Control, Speedup, Efficiency and Through, Nonlinear Pipeline Processors: Reservation and Latency Analysis, Collision Free Scheduling, Pipeline Schedule Optimization. Multiprocessor: Loosely Coupled Multiprocessors, Tightly Coupled Multiprocessors

---

Internal Continuous Assessment (ICA):

In tutorial sessions, students of different batches should be assigned exercise problems and should be guided for the solution. Minimum one tutorial per Unit is expected.

ICA shall consist of minimum ten assignments from the below list.

1. Write an IAS program to compute the results of the following equation. a. Y=ΣXNx=1
   Assume that the computation does not result in an arithmetic overflow and X,Y,and N are positive integers with N≥1.Note: The IAS did not have assembly language, only machine language.
   Use the equation Sum(Y)=(N(N+1))/2 when writing the IAS program.
   Do it the "hard way" without using the equation from part (a).
2. Consider a hypothetical 16 bit microprocessor having 16 bit instructions composed of two fields. The first 4 bits contain the opcode and the remainder the immediate operand or an operand address.
   a. How many different opcodes are there at most?
   b. What is the maximum directly addressable memory capacity (in bytes)
   c. How many bits are needed for the program counter and the instruction register?
3. Propose non restoring algorithm, compare it with restoring algorithm with example.
4. Express the following numbers in IEEE 32-bit floating point format:
   a) -5 b) -6 c)-1.5 d) 384 e) 1/16 f) -1/32
5. Compare zero, one, two and three address machine by writing programs to compute X= (A+B)x(C+D)
6. For a direct mapped cache ,a main memory address is viewed as consisting of three fields. List and define the three field. For an associative cache ,a main memory address is viewed as consisting of two fields. List and define the two fields. For set associative cache , a main memory address is viewed as consisting of three fields. List and define the three fields.
7. Differentiate the different memory replacement policies (FIFO,LRU,OPT and Stack) with examples.
8. Discuss the two types of errors that can occur in a semiconductor memory system.
9. List out the differences between internal memory and external memory.
10. What are the sequences of steps involved in the control of the transfer of data from an external device to the processor?
11. Discuss the drawback of DMA that initiated the need of a new and more efficient techniques known as direct cache access(DCA).
12. Differentiate between hardware control and multiplier control.
13. Differentiate Linear Pipeline Processors and Nonlinear Pipeline Processors.
14. Differentiate Loosely Coupled Multiprocessors and Tightly Coupled Multiprocessors

Text Books:

1. Computer Organization and Architecture-Designing for Performance-William Stallings-Tenth Edition- Pearson
2. Computer Architecture and Organization-John P.Hayes-Third Edition-Tata McGraw-Hill Edition
3. Advanced Computer Architecture-Parallelism,Scalability,Programmability-KaiHwang-Tata McGraw- Hill Edition.

Reference Books:

1. Computer Architecture and Parallel Processing-Kai Hwang and Faye A.Briggs-McGraw-Hill Edition.
2. Computer Organization- V. Carl Hamacher ,Zvonko G. Vranesic ,Safwat G. Zaky-McGraw-Hill Edition.
3. Computer Organization and Architecture-Alan Clements-Cengage Learning.

Solapur University, Solapur
T.E. (Computer Science & Engineering)
Semester I
CS316 - JAVA Programming

Teaching Scheme	**Examination Scheme**
Theory: 3 Hrs/Week	ICA: 25 Marks
Practical: 4 Hrs/Week	POE: 50 marks

Introduction :

This course introduces Java Programming from basics to advanced Java concepts. The importance of Java language cannot be denied as it has already started ruling over the entire Software Industry. The aim of this course is to provide students with an understanding of the object-oriented design and programming techniques. Java, a prime object-oriented programming language, is used to illustrate this programming paradigm

---

COURSE OBJECTIVES:

1. To learn Object oriented programming paradigms using Java language.
2. To introduce the 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 OUTCOME:

Students 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 of Java and Strings in Java (5 hrs)

Basics: Java Runtime Environment, Naming Conventions, Languages Basics: Variables, Operators, Expressions, Statements, Blocks, Control flow Statements, Input and Output, Data Types, Arrays

Fundamentals: String Class and Methods, Immutability of Strings, String Buffer Class and Methods, String Builder class and Methods

Unit 2: Introduction to OOPs (6 hrs)

Objects and Classes, Fields and Methods, Abstraction, Encapsulation, Extending Classes and Inheritance, Types of Inheritance in Java, Polymorphism, Type Compatibility and Conversion, Overriding and Hiding Methods, Hiding Fields, Using the Keyword “super”, Access control, Modifiers, Constructors, Overloading methods, Abstract classes, Nested classes, Packages, Wrapper classes, Int erfaces, Using the Keyword “this” , Object Life time & Garbage Collection, Recursion in Java, Type Casting.

Unit 3: Exceptions, Error Handling and Basic IO (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 , Custom Exceptions, 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 (6 hrs)

Introduction, The Arrays Class, Searching and sorting arrays of primitive data types, Sorting Arrays of Objects, The Comparable and Comparator Interfaces, Sorting using Comparable & Comparator, Collections: Lists, Sets, Maps, Trees, Iterators and Collections, The Collection Class.

SECTION-II

Unit 5: Multithreading and Network Programming (6 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: GUI Programming with AWT and Swing (6 hrs)

Hierarchy of classes in AWT and Swing package, Layouts, Events, Listeners and Event handling, AWT and Swing Components.

Unit 7: JDBC and RMI (6 hrs)

JDBC: Introduction to JDBC, JDBC Drivers & Architecture, CRUD operations Using JDBC API.

RMI: Introduction, RMI Architecture, The Remote Interface, The Remote Object, Writing the Server and Client, Remote Methods, Arguments and Return Values, Stub and Skeleton Classes.

Unit 8: Servlets and JSP (5 hrs)

Introduction to Servlets and JSP, Servlet architecture and lifecycle, JSP architecture and lifecycle. JSP Elements. Requests and Response Objects in Servlet API and JSP, Cookies and Session Handling using Servlet API and JSP.

Internal Continuous Assessment (ICA) :

- Students should undertake minimum of 10 to 15 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 recommend 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.

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
4. Core Java for Beginners- Rashmi Kanta Das, Vikas Publishing House Pvt Ltd.

Reference Books:

1. The Java Language Specification, Java SE 7 Edition Book by James Gosling, Oracle Inc. (e-Resource: http://docs.oracle.com/javase/specs/)
2. Java: The Complete Reference 8 Edition - Herbert Schildt , Tata McGraw - Hill Education
3. The Java™ Tutorials. Oracle Inc. (e-Resource: http://docs.oracle.com/javase/tutorial/)
4. Java Server Programming for Professionals - Ivan Bayross, Sharanam Shah, Cynthia Bayross and Vaishali Shah, Shroff Publishers and Distributors Pvt. Ltd, 2nd Edition

Solapur University, Solapur
T.E. (Computer Science & Engineering)
Semester II
CS321 - COMPILER CONSTRUCTION

Teaching Scheme	**Examination Scheme**
Lectures: 4 Hrs/week, 4 Credits	ESE: 70 Marks
Practical: 2Hrs/week, 1 Credit	ISE: 30 Marks
	ICA: 25 Marks

Introduction:

A compiler translates a program written in a high-level programming language that is suitable for human programmers into the low-level machine language that is required by computers.

Since writing a compiler is a nontrivial task, it is a good idea to split the compilation into several phases with well-defined interfaces. Conceptually, these phases operate in sequence, each phase except first phase taking the output from the previous phase as its input. Each phase is handled by a separate module.

This course provides an in-depth view of translation and optimization process. All phases required for translating a high-level language to machine language is covered in this course including scanning, parsing, intermediate-code generation, machine-code generation, register allocation and code optimization.

Course Prerequisite:

- Theory of Computation.
- System Programming
- Programming Language knowledge

COURSE OBJECTIVES:

1. To introduce the different phases involved in compilation process.
2. To introduce methodologies required for language translation.
3. To design various phases of compiler.
4. To simulate the different phases of a compiler through use of programming languages and tools

---

COURSE OUTCOMES:

At the end of the course, students will be able to

1. Demonstrate various phases involved in compilation process.
2. Differentiate between methodologies required for language translation.
3. Design various phases of compiler.
4. Implement various phases of compiler through use of programming languages and tools.

Course Instructions:

- Use of Free Open source software - YASS to better understand each phase of compiler
- Use of Free Open source software - Parsing Emulator and JFLAP to better understand the parsing technique

SECTION-I

Unit 1 Introduction to Compiling (3 Hrs)

Introduction, Compilers, Phases of a compiler, Compiler construction tools, A simple one pass compiler

Unit 2 Lexical Analysis (7 Hrs)

Role of a Lexical analyzer, Input buffering, Specification and recognition of tokens, Finite automata implications, Designing a lexical analyzer generator

Unit 3 Syntax Analysis **(10 Hrs)**

Role of Parser, Writing grammars for context free environments, Top-down parsing, Recursive descent and predictive parsers (LL), Bottom-Up parsing, Operator precedence parsing, LR parsers, SLR parsers, LALR parsers.

Unit 4 Syntax Directed Translation **(8 Hrs)**

Syntax directed definitions, construction of syntax tree, Bottom-up evaluation of S-attributed definitions, L-attributed definitions, Top-down translation of inherited attributes, Bottom-up evaluation of inherited attributes, Analysis of syntax directed definitions.

SECTION-II

Unit 5 Run Time Environments **(5 Hrs)**

Source language issues, storage organization and allocation strategies, Parameter passing, Symbol table organizations and generations, Dynamic storage allocations

Unit 6 Intermediate Code Generation **(7 Hrs)**

Intermediate languages, declarations, Assignment statements, Boolean expressions, case statements Back patching, procedure calls, Back patching, procedure calls

Unit 7 Code Generation **(8 Hrs)**

Issues in design of a code generator and target machine, Run time storage management, Basic blocks and flow graphs, Next use information and simple code generator, Issues of register allocation, Assignment and basic blocks, Code generation from DAG and the dynamic code generation algorithm

Unit 8 Code Optimization **(8 Hrs)**

Sources of optimization, Peephole optimization and basic blocks, loops in flow graphs, Data flow analysis and equations, code improving transformation and aliases, Data flow analysis and algorithms, symbolic debugging of optimized code

---

Internal Continuous Assessment (ICA) :

1. Generate the grammar for ‘C’ language.
2. Implement the lexical analyzer for simple ‘C’ language.
3. Implement the recognizer for given transition diagram.
4. Implement the top-down parsing using recursive decent parsing technique.
5. Implement the shift- reduce parser.
6. Implement the operator precedence parser.
7. Implement the LL(1) parser for the language.
8. Generate the symbol table for language.
9. Generation of 3- address code for language.
10. Implement the code optimization technique on the code produced in 10.
11. Generation of target code for given 3-address code.
12. Use of free open source software to practice the parsing example.

Text Books:

1. Compilers - Principles, Techniques and Tools - A.V. Aho, R. Shethi and J.D. Ullman (Pearson Education.)

References :

1. Compiler Construction - Dhamdere (Mc-Millan)
2. Compiler Construction – Principles & Practice – Ken Louden (Cengage Learning)
3. Compiler Design in C – Allen I. Holub (PHI / Pearson Education)
4. Compiler Construction: An advance course- Manish Kumar Jha (Dhanpat Rai) (3rd Edition)

Solapur University, Solapur
T.E. (Computer Science & Engineering)
Semester II
CS322–UNIX OPERATING SYSTEM

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hrs/week, 3 Credits	ESE: 70 Marks
Practical: 2 Hrs/week, 1 Credit	ISE: 30 Marks
	ICA: 25 Marks

Introduction:

Unix is the first & the foremost operating system which became very popular worldwide. A multitude of operating systems are prevalent in today's world, but all these modern-day operating systems are evolved & inspired from Unix operating system only. This course primarily deals with the significant basic concepts including architecture of Unix, buffer cache, file subsystem & their understanding through algorithms.

On top of that this course ensures understanding of system calls and their utilities with the help of algorithms. Process control subsystem which includes structure of process, process control & process scheduling is also an integral part of this course. Further it ensures the knowledge of important responsibilities of operating system such as memory management & I/O subsystem. With this kind of a knowledge base of concepts of Unix OS, a student will be able to understand & apply learned concepts while designing / customizing operating system according to specific needs of the world.

Course Prerequisite:

Prerequisite knowledge of Operating Systems Concepts and 'C' programming language is needed.

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.
2. To understand the concept of retrieval of buffer.
3. To explain the structure of regular file (Inode assignment, Disk blocks)
4. To provide hand on system calls related to File System.
5. To introduce fundamentals memory management and I/O management.
6. To provide hands on commands of UNIX and Shell Programming.

COURSE OUTCOMES:

At the end of this course, the students will be able to,

1. Explain the structure of Unix.
2. Implement scenarios for retrieval of buffer.
3. Implement algorithms for inode assignment and disk block allocation. (Regular File)
4. Write syntax for different system calls and use these system calls in program.
5. Explain the concepts of Memory and I/O management.
6. Write and execute programs using shell programming.

SECTION–I

Unit-1:Introduction **(6 Hrs.)**

General Overview of the System- History, System Structure, User Perspective, Operating System Services, Assumption about Hardware Introduction to KERNEL- Architecture of UNIX OS, Introduction to System Concepts, Kernel data structure, System Administration.

Unit-2: The Buffer Cache **(5 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 and System Calls (7 **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. System call: File operations, Pipe, mount – unmount and Link – unlink.

SECTION–II

Unit-4:The Structure of process **(5 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-5:Process Control and Scheduling **(8 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, process scheduling, system call for time, clock.

Unit-6:Memory Management Policies **(5 Hrs.)**
Swapping, Demand passing, a hybrid system with demand paging and swapping

Unit-7: The I/O Sub **(4 Hrs.)**
Driver interfaces, disk drives, terminal drivers, Streams.

Internal Continuous Assessment (ICA) :
It should consist of minimum 8-10 experiments based on above topics. Following experiments may be conducted for the term work.

Practical List:

1. Study of Unix architecture
2. Write a program to implement cp command
3. Write a program to implement *ls* command
4. Write a program to implement *getblk* algorithm
5. Write a program to implement *ialloc* & *ifree* algorithm.
6. Write a Program to implement *alloc* and free algorithm.
7. Study of System calls
   - Write a program to implement STAT & FSTAT,
   - Write a program to implement PIPES,
   - Write a program to implement LINK &UNLINK,
   - Write a program to implement DUP,
   - Write program to implement MOUNT&UNMOUNT.
8. 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 fibonnaci series
   - WAP to find sum of series of entered number
   - WAP to find power of number.
9. WAP to implement *malloc* algorithm.
10. Study of KERNEL module programming

Text Books:

1. The design of Unix Operating Systems- Maurice J. Bach(PHI)
2. Unix Manual

Reference Books:

1. Unix concepts and administration – 3rd Edition- Sumitabha Das (TMGH).
2. Advanced Programming in the UNIX Environment by W. Richard Steven.
3. UNIX Concepts & Applications by Sumitabha Das

Solapur University, Solapur
T.E. (Computer Science and Engineering)
Semester II
CS323 - MOBILE COMPUTING

Teaching Scheme	**Examination Scheme**
Lecture: 3 Hrs/Week	Theory: 70 Marks
Tutorial: 1 Hr/Week	ISE: 30 Marks
	ICA: 25 Marks

Introduction:
Mobile Computing is a technology that allows transmission of data, voice and video via a computer or any other wireless enabled device without having to be connected to a fixed physical link. This subject will give an overview of Mobile Computing and then it will take you through how it evolved and where is the technology headed to in future along with the GSM & GPRS system.

COURSE OBJECTIVES:
1. To introduce concepts and principles of mobile computing.
2. To explore skills of finding solutions for mobile computing applications.

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.

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

SECTION-I

Unit 1: **Introduction to Wireless Communication** **(5 Hrs.)**
History of wireless communication, Applications, Generations: 1G, 2G, 3G and 4G, Modulation: Digital Modulations & Analog Modulation, Demodulation

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

Unit 3: Medium Access Control **(6 Hrs.)**
Need of MAC algorithms, SDMA, FDMA, TDMA, CDMA.

Unit 4: GSM **(6 Hrs.)**
Mobile services, System architecture, Radio interface, Protocols, Localization and calling, Handover, Security.

SECTION II

Unit 5: New Data Services in GSM **(5 Hrs.)**
HSCSD, General packet radio service(GPRS) - GPRS and packet data network, GPRS architecture, GPRS transmission plane protocol reference model, Applications of GPRS, Limitations of GPRS, Mobile Number Portability

Unit 6: Wireless LAN **(6 Hrs.)**
Introduction, advantages and design goals for wireless LAN, Infrastructure, ad-hoc networks, IEEE 802.11: system and protocol architecture, physical layer, HIPERLAN protocol architecture and physical layer and MAC, Bluetooth physical and MAC layer, Introduction of Wireless ad-hoc networks.

Unit 7: Mobile Network Layer **(6 Hrs.)**
Mobile IP, DHCP

Unit 8: Mobile Transport Layer **(6 Hrs.)**
Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, Fast & Selective retransmission & Recovery, Transmission Oriented TCP

Internal Continuous Assessment (ICA) :
Assignment List:

1. The message signal x(t)=Sin(100t) modulates the carrier signal c(t)=A Cos(2πfct).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 hand over, 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 GPRS work? What are the data services used in GPRS?
7. How do IEEE 802.11,HiperLAN2 and Bluetooth, respectively, solve the hidden terminal problems?
8. 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?
9. What is the basic purpose of DHCP? Name the entities of DHCP. How can DHCP be used for mobility and support of mobile IP?
10. How and why does I-TCP (Indirect TCP) isolate problems on the wireless link? What are the main drawbacks of this solution?

Text Books:

1. Mobile Communications – Jochen Schiller (PEARSON) (Chapters: 1, 2, 3, 5, 7, 8)
2. Introduction to Wireless and Mobile System-D.P.Agrawal and Qing-AnZeng (CENGAGE) (Chapter:1,6)

Reference Books:

1. Wireless Communication –Principles and practice - Theodore S. Rappaport (PEARSON)
2. Mobile and Personal Communication Systems and Services - Raj Pandya –(PHI)
3. Mobile Computing-Technology, Applications and Service Creation-Asoke K Talukder, Hasan Ahmed and Roopa R Yavagal.(MGH)

Solapur University, Solapur
T.E. (Computer Science and Engineering)
Semester II
CS324 - SOFTWARE ENGINEERING

Teaching Scheme	**Examination Scheme**
Lecture: 3 Hrs/week	ISE: 30 Marks
Tutorials: 1 Hr/week	ESE: 70 Marks
	ICA: 25 Marks

COURSE OBJECTIVES :

The Course should enable the student

1. To help students to develop skills that will enable them to develop software of high quality which is reliable, easy to understand, modify and maintain.
2. To illustrate and compare use of life cycl 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.
5. To implement the risk management system in software project.

COURSE OUTCOMES :

The Student should able to

1. Students will be able to develop the software project using appropriate process.
2. Develop a Software Project from requirement gathering to implementation.
3. Develop design schemes in software project.
4. Estimate the cost and effort of software project.
5. To improve the quality of the software project.
6. Identify the impact of risk in software development life cycle.

SECTION-I

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

SDLC Definition, Phased Development Process, Software processes, Characteristics of a software process, Software Development Process Models : Waterfall model, Prototype model, Iterative model, Spiral model, Verification and validation model. Introduction to Agile Project Management

Unit 2: Software Requirements Analysis and Specification (4 Hrs.)

Definition of Software Requirements, Need of SRS, Requirement process, Problem Analysis, Requirements Specification: Characteristics of SRS , Components of SRS and general structure of SRS.

Unit 3: Function and Object Oriented Design (8 Hrs.)

Design Principles, Module Level Concepts, Design Notation and Specification, Structured Design Methodology. OO Analysis and OO design, Concepts, Design Notation and Specification, Design Methodology, UML Diagrams.

Unit 4: Testing (5 Hrs.)

Testing Fundamentals, Black box and White box testing, Testing Process
Introduction to types of testing : system testing, load testing, stress testing, Integration Testing, Unit testing , User Acceptance Testing white box and black box testing, Regression testing, sanity testing, smoke testing.

SECTION-II

Unit 5: Project Management Process (2 Hrs.)

Overview: Project Management Process, Software Configuration Management process, Process Management Process.

Unit 6: The Project Planning (5 Hrs.)

The Project Planning Infrastructure-Process Database, Process Capability Baseline, Process, Asset and the body of knowledge system, Requirement Change management, Effort Estimation and Scheduling –Concepts, Effort estimation, Scheduling.

Unit 7: Quality planning , RiskManagement and Tracking (5 Hrs.)

Quality Concepts, Qualitative Quality Management Planning, Defect Prevention Planning, Concepts of Risk and Risk Management Assessment, Risk Control, Concepts in Measurements: Measurements, Project Tracking

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

Processes and Project Management, Project Management and The CMM, Team Management, Customer Communication and Issue Resolution, The structure of the Project Management Plan, Concepts in Configuration Management, Configuration Management Process, Reviews Process, Milestone Analysis, Defect Analysis and Prevention, Project Closure Analysis.

Internal Continuous Assessment (ICA) :

Implémentation of mini software project by applying SDLC cycles.
It should consist of minimum 6 - 8 assignments based on each topic of above syllabus.

Text Books:

1. An Integrated Approach to Software Engineering- 3rd edition: Pankaj Jalote(Narosa Publishers)
2. Software Project management in practice-Pankaj Jalote

References:

1. Effective Project Management Traditional,Agile,Extreme ,Robert K. Wysocki WILEY INDIA,6th edition.
2. Ian Sommerville,software engineering, pearson education Asia, 6th edition
3. Software Engineering Fundamentals –Ali Behforooz and Frederick j. Hudson (Oxford University Press)
4. PANKAj JALOTE'S Software Engineering, A Precise Approach(Wiley Precise Textbook, WILEY INDIA)
5. Software Engineering- Practioner Approach: Roger S. Pressman. 6th edition.

Solapur University, Solapur
T.E. (COMPUTER SCIENCE & ENGINEERING)
Semester II
CS325 - MOBILE APPLICATION DEVELOPMENT

Teaching Scheme	**Examination Scheme**
Lecture: 3 Hours/week	Theory: 70 Marks
Practical: 2 Hours/week	ICA: 25 marks
	POE: 50 marks

Introduction :
This course introduces Mobile Application Development presenting various approaches and technologies to build mobile apps. The course covers several facets of composing mobile apps-design, development, validation, packaging and distribution of apps.

Prerequisite(s): Basic understanding of Java Programming Language

---

COURSE OBJECTIVES:

1. Develop mobile applications using modern mobile development tools for android.
2. Independently manage all phases of mobile project development.
3. Develop applications that effectively combine mobile device capabilities such as communication, computing.

COURSE OUTCOMES:

1. Familiarize with mobile apps development aspects.
2. Design & develop mobile apps, using Android as a development platform.
3. Perform testing, signing, packaging and distribution of mobile apps.

SECTION–I

Unit 1: Android Operating System (5 Hrs.)
Introduction, Mobility Panorama, What is Android, History of Android, Why develop for Android, Android Features and Applications, API Level, App Development Approaches, Android Architecture.

Unit 2: Getting started with Mobility (6 Hrs.)
Configuring Android development environment, Downloading and installing the Android Studio, Downloading and updating the Android SDK. Mobility landscape, Mobile platforms, Mobile apps development, Overview of Android platform, setting up the mobile app development environment along with an emulator, a case study on Mobile app development, Logical Components of an Android App, Android tool Repository.

Unit 3: User Interface Design. (7 Hrs.)
Activity, Activity states, Activity Life Cycle, UI Resources, Layout Resources, String Resources, Image Resources, UI Elements and Events, Interaction between Activities, Exchanging data among activity, Fragments, Lice Cycle of Fragments, Interaction between Fragments.

Unit 4 : Mobile Application Functionality- Beyond UI (5 Hrs.)
App functionality beyond user interface - Threads, Async task, Services – states and life cycle, Intent and Bound Service, Notifications, Intents and Intent Resolution, Broadcast receivers, Telephony and SMS APIs

SECTION–II

Unit 5: Native data handling (6 Hrs.)

On-device File I/O, data persistent and access using shared preferences, mobile databases such as SQLite and implementation for CRUD, and enterprise data access (via Internet/Intranet)

Unit 6: Sprucing up mobile apps (6 Hrs.)

Android Graphics: Supporting Multiple Screens, Drawables, Custom View and Canvas, Android Animation: Drawable Animation, View Animation, Property Animation, Multimedia: Audio, Video and images, Media Player API, Location Services, Maps, and native hardware `access (sensors such as accelerometer and gyroscope)

Unit 7: Testing mobile apps (5 Hrs.)

Debugging mobile apps, White box testing, Black box testing, and test automation of mobile apps, JUnit for Android, Robotium, MonkeyTalk

Unit 8: Taking apps to Market (5 Hrs.)

Versioning, signing and packaging mobile apps, distributing apps on mobile market place, Google play store.

List of Assignments:

1) Objective of assignments should be to test students understanding and asses their ability to put into practice the concepts and terminologies learned.
2) Assignments must be of nature which requires students to identify the use case scenarios for developing Mobile Apps based on mentioned syllabus.
3) 10-15 assignments on the above syllabus.

Text Books:

1. "Android Application Development All in one for Dummies" by Barry Burd
2. "Mobile Apps Development" by Anubhav Pradhan, Anil V Deshpande
3. "Embedded Android-Porting, Extending, and Customizing" by Karim Yaghmour (O'Reilly Media)

Reference Books:

1. Android Developer Resources: http://developer.android.com
2. Android Developer Tools Essentials by Mike Wolfson (O'Reilly Media).

Solapur University, Solapur
T.E. (Computer Science and Engineering)
Semester II
CS326 - PROGRAMMING IN C#.NET

Teaching Scheme	**Examination Scheme**
Lectures – 3 Hours/week, 2 Credits	POE – 50 Marks
Laboratory – 2 Hour/week, 1 Credit	ICA – 25 Marks

Introduction :

This course introduces C# programming language. C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

Course Prerequisite :

C# programming is very much based on C and C++ programming languages, Students must have a basic understanding of C or C++ programming, then it will be fun to learn C#.

COURSE OBJECTIVES :

1. To learn .Net Framework
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 implement object oriented concepts using C#.
4. To develop database applications using ADO.Net
5. To develop ability to design and build Object Oriented and GUI, Web applications on Windows Platform.

COURSE OUTCOMES:

Upon successful completion of this course, students will be able to:

1. Knowledge of the structure and model of the programming language C # (note)
2. Use the programming language C # for various programming technologies (understanding)
3. Develop software in C # (application)
4. Evaluate user requirements for software functionality required to decide whether the programming language C # can meet user requirements (analysis)
5. Propose the use of certain technologies by implementing them in the C # programming language to solve the given problem (synthesis)
6. Choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (evaluation)

SECTION-I

Unit-1 : Introduction to .NET Framework (03 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 IDE.

Unit-2: C# Application Basics and Language fundamentals (04 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 Concepts and Exception Handling using C# (06 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, Need for Exceptions, Exception Hierarchy, Handling Exceptions using try-catch-finally blocks, creating and defining Custom Exceptions, the "throw" keyword.

Unit-4 : Delegates Events andMultithreading (04 Hrs.)
Events and Delegates in C#, Multicast Delegate, Event Handling What is Multithreading, Multithreading in C#, Static and Instances members of Thread Class, Basic Thread operations, Thread priorities, Thread Synchronization.

SECTION II

Unit-5 : File System and Streams, Generics (04 Hrs.)
Streams and System.IO namespace, Console IO, Reading writing and updating files and directories, System.IO.FileInfo Class, Serialization and Deserialization.
Generics: Introduction to Generics, Benefits of Generics, Generic Type Parameters, Constraints on Type Parameters, Generic Classes, Generic Interfaces, Generic Methods, Generics and Arrays Generic Delegates/

Unit-6 : GUI Programming in C# (04 Hrs.)
Windows Forms and System.Windows.Form namespace, Building Windows Forms Applications using IDE, Windows Form controls , Event Handling , List Box , Combo Box, Tree View, File Dialog, Tool Bar, Windows standard Dialog Boxes, Menu Bar.

Unit-7 : Data access using ADO.NET (04 Hrs.)
Introduction to ADO.NET, System.Data namespace, DataSet, DataTable, DataRow, DataColumn and other prominent classes, Accessing and Updating Data using ADO.NET.

Unit-8 : Introduction ASP.NET (06 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#.

Outcome Assessment Strategies and ICA:
Students will be expected to successfully complete a group of C# projects. The projects will include the following:

1. Windows application using windows controls and events
2. Web application (ASP.NET)
3. ADO.NET database application
4. Building and using classes, events, methods, properties.

Text Books:

1. Professional C#, 3rd Edition -Simon Robinson, Christian Nagel, Karli Watson, Jay Glynn, Morgan Skinner, Bill Evjen, Wrox Press - Wiley India.
2. Programming in C#: A Primer 3 Edition -E Balagurusamy, Tata McGraw - Hill Education

Reference Books:

1. C# Language Specification Version 5.0 Microsoft. (E-Resource available at http://www.microsoft.com)
2. C# Programming Guide MSDN, Microsoft. (http://msdn.microsoft.com/en-US/) 3) Microsoft Visual C# Step by Step 2010 - John Sharp, Microsoft Press.

SLH 32 - Self Learning (Technical) – 1. Computer Modeling and Simulation

Examination Scheme
Theory: 50 marks

COURSE OBJECTIVES:

1. To learn fundamentals of Computer Modeling and Simulation.
2. To study and apply Network Simulator software.
3. To learn the basics of writing code for simple scenarios in Network Simulator.

COURSE OUTCOME:

1. Students are able to understand Modeling and Simulation.
2. Students are able to apply and explore the basic API libraries for Network Simulator.
3. Students are introduced to the network modeling and simulation using 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:

1. Discrete-Event System Simulation–Jerry Banks, John S. Carson II, Barry L. Nelson,David M. Nicol
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

SLH 32 -Self Learning (Technical) – 2. Software Licences and Practices

Examination Scheme
Theory: 50 marks

COURSE OBJECTIVES:

1. To educate students about various software licensing models and practices
2. To develop first hand information about various software licenses adopted in open source and non open source software development.

COURSE OUTCOMES:

1. Students will develop basic understanding of software licensing models and practices adopted in software development and distribution.
2. Students will be able to analyze and choose appropriate software licensing model and strategy for their own softwares developed.

Unit 1: Introduction

Basic Principles of Copyright Law, Contract and Copyright, Open Source Software Licensing, Issues, with Copyrights and Patents, The Open Source Definition, Warranties.

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

Entering Contracts, Statutory Developments Related to Software Contracts, The Self Enforcing Nature of Open Source and Free Software Licenses, The Global Scope of Open Source and Free Software Licensing, The "Negative Effects" of Open Source and Free Software Licensing, CommUnity Enforcement of Open Source and Free Software Licenses, Compatible and Incompatible Licensing: Multiple and Cross Licensing.

Textbooks:

Understanding Open Source and Free Software Licensing - By Andrew M. St. Laurent, Oreily Media. (e-Resource available at : http://oreilly.com/openbook/osfreesoft/book/index.html)

Reference Books:

1. Intellectual Property and Open Source: A Practical Guide to Protecting Code - By Van Lindberg, Oreily Media.
2. Essentials of Licensing Intellectual Property - By Alexander I. Poltorak and Paul J. Lerner, John Wiley Publication.

Solapur University, Solapur
T.E. (Computer Science and Engineering)
Semester - II

SLH 32 - Self Learning (Technical) – 3. Network Setup and Management Tools
Examination Scheme
Theory: 50 marks

COURSE OBJECTIVES:

At the end of the course the student will understand:

1. Computer Network Setup requirements.
2. Five major functional areas of network management namely Configuration and Fault management, Accounting, Performance and Security Management.
3. Easy ways for network setup and management tools.

COURSE OUTCOME:

1. Students are exposed to understand How to Setup and Computer Network.
2. Students are able to understand and use the network management tools.
3. Students get aware with network management tools.

Unit 1: Computer Network Setup

Network Interface Adapters: NIC Functions, Features, Selecting a NIC, Network Connection Devices: Hubs, Switches, VLAN, Layer-3 Switches, Designing a Network: Network design overview, Designing an Internetwork.

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

Fault Management, Fault identification and isolation, Event correlation Techniques, Security Management, Host and User Authentication, Key Management.

Unit 4: Management Tools, Systems and Applications

Testing and Monitoring Tools, Integrating Tools, Development Tools, Web-based Enterprise Management.

Books and References:

1. Networking – The Complete Reference by Craig Zacker Tata McGraw Hill (Unit 1)
2. Network Management : Principles and Practices by Subramanian M. MA: Addison – Wesley (2000) (Unit 2,3,4)

Solapur University, Solapur
T.E. (Computer Science and Engineering)
Semester - II

SLH 32 - Self Learning (Technical) – 4. Ethical Hacking
Examination Scheme
Theory: 50 marks

COURSE OBJECTIVES:

1. To know what are the different types of attacks.
2. To know various tools to prevent data from attacker.

COURSE OUTCOMES:

Upon completion of this course, the student should be able to

1. Defend hackers attacks and protect data
2. Defend a computer against a variety of security attacks using various tools

---

Unit 1: Introduction to Hacking

Introduction to Hacking, Importance of Security, Elements of Security, Phases of an Attack, Types of Hacker Attacks, Legal implication of hacking, Impact of hacking. Introduction to Footprinting, Footprinting Tools.

Unit 2: Scanning and Enumeration

Introduction to Scanning, Objectives, Scanning Methodology, Tools, Introduction to Enumeration, Enumeration Techniques, Enumeration Procedure, Tools.

Unit 3: Denial of Service and Session Hijacking

Denial of Service, Types of DoS Attacks, DDoS Attacks, Smurf Attack, SYN Flooding, DoS/DDoS Countermeasures, Session Hijacking, Spoofing vs. Hijacking, Types of Session Hijacking, Session Hijacking Steps, Prevention of Session Hijackinng.

Unit 4: Penetration Testing

Introduction, Security Assessments, Types of Penetration Testing, Phases of Penetration Testing, Tools , Penetration Testing Tools.

Books and References:

1. Ec-Council, “Ethical Hacking and Countermeasures: Attack Phases”, Delmar Cengage Learning, 2009.
2. Michael T. Simpson, Kent Backman, James E. Corley, “Hands-On Ethical Hacking and Network Defense”, Cengage Learning, 2012.
3. Patrick Engebretson, “The Basics of Hacking and Penetration Testing – Ethical Hacking and Penetration Testing Made Easy”, Syngress Media, Second Revised Edition, 2013.

Introduction

Data science is a field of study and application that has been growing rapidly for the past several decades. As a growing field, it is gaining a lot of attention in both the media as well as in the job market. This course introduces the basic terminology used by data scientists and a look at the types of problem.

Course Prerequisite(s): Basic terminologies of Mathematical fundamentals.

COURSE OBJECTIVES:

1. To introduce the basic terminology used by data scientists
2. To explain steps of data science and types of data science
3. To use visualizations in order to share results in communicable form

---

COURSE OUTCOME:

Students will be able to

1. elaborate the basics of data science and its applications
2. classify various types of data science
3. visualize the data in multiple forms

SECTION-I

Unit 1: Introduction to Data Science and Applications

What is data science?, Basic terminology, Why data science?, Example – Sigma Technologies, The data science Venn diagram, the math Example – spawner-recruit models, computer programming, why Python?, Python practices, Example of basic Python, Example – parsing a single tweet, Domain knowledge, Data science case studies, Case study – automating government paper pushing Fire all humans, right?, Case study – marketing dollars Case study – what's in a job description?

Unit 2: Types of Data

Flavors of data, structured versus unstructured data, example of data preprocessing, word/phrase counts, Presence of certain special characters Relative length of text Picking out topics, quantitative versus qualitative data, examples. The four levels of data, **the nominal level:** Mathematical operations allowed, measures of center, **the ordinal level:** Examples, mathematical operations allowed, measures of center, **the interval level**: Examples, mathematical operations allowed, measures of center, measures of variation, standard deviation, **the ratio level**: Examples, measures of center, problems with the ratio level

SECTION-II

Unit 3: The Five Steps of Data Science

Introduction, overview of the five steps: ask an interesting question, obtain the data, explore the data, model the data, communicate and visualize the results, Explore the data: basic questions for data exploration, examples.

Unit 4: Communicating Data

Why does communication matter? Identifying effective and ineffective visualizations, scatter plots, line graphs, bar charts, histograms, box plots. Graphs and statistics: correlation versus causation, simpson's paradox, verbal communication, the why/how/what strategy of presenting

Text Books:

1. Principles of Data Science by Sinan Ozdemir (2016) Packt Publishing
2. Data science from scratch: first principles with python. by Grus, J. (2015) "O'Reilly Media, Inc."

Reference Books:

1. Data science for dummies by Pierson, L. (2015), John Wiley & Sons Publishing,
2. Hands-On Data Science and Python Machine Learning by Kane, F. (2017), Packt Publishing

Solapur University, Solapur
T.E. (Computer Science and Engineering)
Semester - II
Self Learning (Technical) – 6. UI Technologies
Examination Scheme
Theory: 50 marks

COURSE PREREQUISITE:
Student should have knowledge of basic programming. They should also have basic knowledge of GUI and its working.

COURSE OBJECTIVES:

1. Inculcate skill necessary to design, develop and style a web-based user Interface.
2. Develop the skill for responsive web design.
3. Develop ability to identify use cases for applying client side scripting UI.
4. Develop skill required to create light weight browser based web application using client side scripting framework

---

COURSE OUTCOMES:

1. Design, develop and apply styling to a web-based application.
2. To be able to design responsive web design.
3. Build efficient and scalable web API and application.
4. Develop lightweight browser based functionalities leveraging client side scripting framework.

SECTION -I

Unit 1 HTML5, CSS3, DHTML
HTML: HTML Review, HTML5 Introduction, Futures of HTML5, New elements in HTML5 (canvas, Media elements, Form elements, Semantic and structural elements,
CSS: CSS Introduction ,CSS Syntax, CSS Id & Class, CSS Styling, CSS3: Selectors, Box Model, Backgrounds and Borders, Image Values and Replaced Content, Text Effects, 2D/3D Transformations, Animations, Multiple Column Layout, User Interface. DHTML

Unit 2– JavaScript and jQuery
Introduction, Understanding of variables, data types, control flow, and basic function usage in JavaScript, Event Handling, JS Built-in Objects
JSON: JavaScript Object and Array Creation Using Literals, JavaScript Objects in Arrays & Arrays in Objects, JSON syntax, JSON Parsers, JSON Data Transfer Between Client and Server, AJAX.
jQuery: jQuery Fundamentals, using jQuery Selectors, Interacting with the DOM, Handling Events

SECTION -II

Unit 3: Responsive Web Design
What is Responsive Design, Responsive Content, HTML for Responsive Sites, CSS for Responsive Sites, Media Queries, Images, Responsive Workflow, Mobile and Beyond, Typography, Navigation and Header Layout.

Unit 4- Bootstrap
Introduction to Bootstrap, Bootstrap CSS, Bootstrap Layout Components, Bootstrap JavaScript Plugins

Text Book

1. Head First HTML5 Programming by Eric Freeman, Elisabeth Robson O'Reilly Media
2. HTML5 and CSS3, 2nd Edition Level Up with Todays Web Technolgoes by Brain P. Hogan – Pragmatic Bookshelf, Second Edition.
3. Designing Next Generation Web Projects with CSS3 by Sandro Paganotti- CreateSpace Independent Publishing Platform.
4. JavaScript, A Geginners Guide, Third Edition by John Pollock-McGraw_Hill Osborne Media.
5. Learning Responsive Web Design by Clarissa Peterson. O'Relly Media.
6. Bootstrap by Jake Spurlock, Published by O'Reilly Media

Reference Book

1. HTML5 Black Book: Covers CSS3, Javascript, XML, XHTML, Ajax, PHP and Jquery by Kogent Learning Solutions Inc.
2. HTML5 Application, Zachary Kessin, O'Reilly, Shroff Publishing and Distributions Pvt. Ltd.
3. Responsive Design Workflow –Stephen N. Hay.