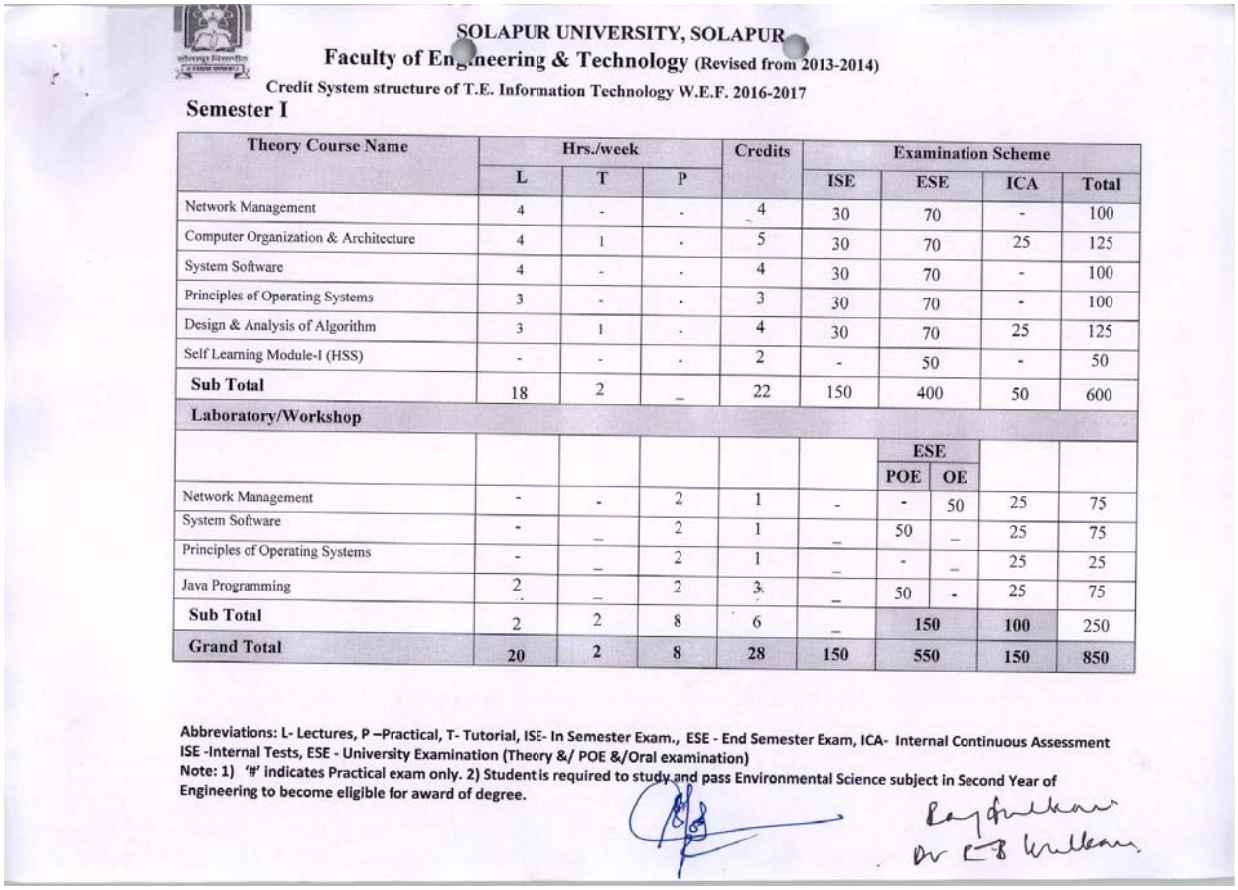
SOLAPUR UNIVERSITY, SOLAPUR

FACULTY OF ENGINEERING & TECHNOLOGY

Structure of T.E. (Information Technology) w. e. f. from 2016-17



SOLAPUR UNIVERSITY, SOLAPUR

Faculty of Engineering & Technology (Revised from 2013-2014)

Credit System structure of T.E. Information Technology W.E.F. 2016-2017

Semester I

Theory Course Name	Hrs./week			Credits	Examination Scheme			
	L	T	P		ISE	ESE	ICA	Total
Network Management	4	-	-	4	30	70	-	100
Computer Organization & Architecture	4	1	-	5	30	70	25	125
System Software	4	-	-	4	30	70	-	100
Principles of Operating Systems	3	-	-	3	30	70	-	100
Design & Analysis of Algorithm	3	1	-	4	30	70	25	125
Self Learning Module-I (HSS)	-	-	-	2	-	50	-	50
Sub Total	18	2	_	22	150	400	50	600
Laboratory/Workshop								
						ESE		
						POE	OE	
Network Management	-	-	2	1	-	-	50	25	75
System Software	-	_	2	1	_	50	_	25	75
Principles of Operating Systems	-	_	2	1	_	-	_	25	25
Java Programming	2	_	2	3	_	50	-	25	75
Sub Total	2	2	8	6	_	150		100	250
Grand Total	**20**	**2**	**8**	**28**	**150**	**550**		**150**	**850**

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. 2) Student is required to study and pass Environmental Science subject in Second Year of Engineering to become eligible for award of degree.

SOLAPUR UNIVERSITY, SOLAPUR.

Faculty of Engineering & Technology (Revised from 2013-2014)

Credit System structure of T.E. Information Technology W.E.F. 2016-2017

Semester II

Theory Course Name	Hrs./week			Credits	Examination Scheme			
	L	T	P		ISE	ESE	ICA	Total
Database Engineering	4	-	-	4	30	70	-	100
Object Oriented Modeling & Design	3	1	-	4	30	70	25	125
Unix Operating System Concepts	4	-	-	4	30	70	-	100
Software Engineering	3	1	-	4	30	70	25	125
Mobile Application Development	4	-	-	4	30	70	25	125
Self Learning Module-II (HSS/Technical)	-	-	-	2	-	50	-	50
Sub Total	18	2	_	22	150	400	75	625
Laboratory/Workshop								
_	_	_	_	_	_	ESE		_	_
						POE	OE		
Database Engineering	_	_	2	1	_	50	_	25	75
Unix Operating System Concepts	_	_	2	1	_	-	25	25	50
Advanced Java	2	_	2	3	_	50	-	25	75
Seminar	-	_	2	1	_	-	_	25	25
Sub Total	2	_	8	6	_	125		100	225
Grand Total	**20**	2	**8**	**28**	**150**	**525**		**175**	**850**

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. 2) Student is required to study and pass Environmental Science subject in Second Year of Engineering to become eligible for award of degree.

Dy R & Tech.

Subjects for Self Learning Module-I: Humanities & Social Science (HSS)

1. Economics
2. Psychology
3. Philosophy
4. Sociology
5. Humanities

Subjects for self learning Module-II: Technical

1. Network setup and management.
2. Tools for Computer Architecture.
3. Compiler Development Tools.

Note:-

- Term work assessment shall be a continuous process based on student's performance in – class tests, assignments, homework, subject seminars, quizzes, laboratory books and their interaction during theory and lab sessions
- The batch size for practical/tutorials shall be of 15 students. On forming the batches, if the strength of remaining students exceeds 7 students, then a new batch may be formed.
- Vocational training (to be evaluated at B.E. Semester -I) of minimum 15 days should be done in vacation in part or whole from S.E. Semester -II to commencement of B.E. Semester – I and the report should be submitted in B.E. Semester -I.
- Student shall select one Self Learning Module at T.E. Part I and T.E. Part II each from Technical and Humanities and Social Sciences Group with at least one Self Learning Module from the Humanities and Social Sciences Group
- Curriculum for Humanities and Social Sciences Self Learning Modules is common for all under graduate programmes of faculty of Engineering and Technology

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. 2) Student is required to study and pass Environmental Science subject in Second Year of Engineering to become eligible for award of degree.

Dy R & Tech.

SOLAPUR UNIVERSITY, SOLAPUR

T. E. (INFORMATION TECHNOLOGY)

Part – I

1. Network Management

Teaching Scheme	**Examination Scheme**
Theory: 4 Hrs/Week	Theory:100 Marks
Practical: 2 Hrs/Week	Term-Work: 25 Marks
	OE : 50 Marks

Course Objectives

1) To get acquainted with various networking protocols.
2) To identify the basic layers and functions provided by TCP/IP protocol
3) To identify monitoring & management protocol (SNMP) based on its features & parameters.

Course Outcomes

At the end of the course

1) Students should be able to identify various networking protocols.
2) Students should get acquainted with the basic layers & functions of TCP/IP protocol.
3) Students should be in a position to implement various network management systems using SNMP.

Section I

Unit 1: Transport Layer (8 Hrs)

Overview of IP Protocol, UDP: process to process communication, user datagram, UDP operations, Use of UDP, TCP: TCP services, TCP features, segment, TCP connection, flow control, error control, congestion control, TCP timers, Client server models.

Unit 2: Application Layer (4 Hrs)

Host configuration: BOOTP, DHCP: Static and Dynamic addressing, allocation, manual and Automatic configuration, packet format.

Unit 3: Domain Name System (DNS), Telnet And FTP (4 Hrs)

DNS: Name Space, Domain Name Space, distribution of name space, DNS in internet, Resolution, DNS massages types of records. TELNET: NVT, embedding, out of band signaling, escape character, mode of operations, FTP and TFTP.

Unit 4: SMTP and WWW (4 Hrs)

E-Mail: Architecture, user agents, MTA, pop3, HTTP: Architecture and protocols.

Section II

Unit 5: SNMP 1 Network Management (4 Hrs)

Managed Network: The History of SNMP Management, Internet Organizations and standards, Internet Documents, The SNMP Model, The Organization Model, System Overview.

Unit 6: SNMPV1 Network Management (5 Hrs)

The Information Model – Introduction, The Structure of Management Information, Managed Objects, Management Information Base. The SNMP Communication Model – The SNMP Architecture, Administrative Model, SNMP Specifications, SNMP Operations, SNMP MIB Group, Functional Model.

Unit 7: Network Management Applications (8 Hrs)

Configuration Management, Fault Management, Performance Management, Event Correlation Techniques, Security Management.

Text Books:

1. TCP/IP Protocol Suite : Behroz A. Forozen.
2. Computer Networks, Prentice Hall, Andrew S. Tanenbaum.
3. Network Management principles and practice – Mani Subramanian (Pearson Edition).

Reference Books :

1. TCP/IP Vol 3. : Client Server Programming & Application – Comer
2. SNMP,SNMPv2,SNMPv3 and RMON 1 and 2 –William Stallings

Term work: Minimum 8-10 Experiments based on following guidelines.

List of assignments:-

1. Configuration of network- Assigning IP Address, Subnet mask, Default Gateway & Testing Basic Connectivity. (All commands related to connectivity and testing)
2. Implementing Client-Server program using Iterative UDP server.
3. Implementing Client-Server program using Iterative TCP server.
4. Implementing Client-Server program using Concurrent TCP server.

(For experiments 2, 3 and 4 socket connectivity and commands related to sockets like BIND need to explain)

5. Study of existing network (e.g. college) and design of any new network.
6. Study of different network topologies using simulation & kits.
7. Study of basic SNMP Commands.
8. Study of Net-SNMP.
9. Study of Request for Comment (RFC) 1157.

SOLAPUR UNIVERSITY, SOLAPUR

T. E. (INFORMATION TECHNOLOGY)

Part– I

2. Computer Organization & Architecture

Teaching Scheme
Theory: - 4 Hrs/Week
Tutorial: - 1 Hr/Week

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

Course Objectives

1) To enable the student to get acquainted with the working of a computer and it's basic principles.
2) To analyze the performance of computer systems and design prototypes of computers.
3) To deal with the issues relating to modern processors like SPARC and use cache and pipelining to improve efficiency.

Course Outcomes

At the end of the course

1) a student should be able to identify the major components of a computer including CPU, memory, I/O and storage.
2) a student should be able to carry on cost performance analysis and design prototypes for computers.
3) a student should be acquainted with assembly language programming of SPARC Processor & use of cache memory.

SECTION-I

Unit 1: Introduction (5 Hrs)

A Brief History of Computing, The Von Neumann Model, The System Bus Model, Levels of Machines, Upward Compatibility, The Levels of computers, A Typical Computer System, Classes of computers, Trends in Computer architecture.

Unit 2: Data Representation and Arithmetic (10 Hrs)

Introduction, Fixed Point Numbers, Floating Point Numbers, Fixed Point Addition and Subtraction, Fixed Point Multiplication and Division, Floating Point Arithmetic, High Performance Arithmetic: High Performance Addition, High Performance Multiplication.

Unit 3: The Instruction Set Architecture (9 Hrs)

Hardware Components of the Instruction Set Architecture, ARC - A RISC Computer , Pseudo-Operations, Synthetic Instructions, Examples of Assembly Language Programs, Accessing Data in Memory-Addressing Modes.

Unit 4: Control Unit Design and Memory (6 Hrs)

Hardwired Control Unit: Design Methods (Sequence counter), Multiplier Control Unit: Introduction, Implementation of Multiplier control unit in each case, The Memory Hierarchy, Cache Memory, Interleaved memory, Associative memory.

SECTION-II

Unit 5: Fundamentals of Pipeline (8 Hrs)

Trends in Power in Integrated Circuits and costs, Dependability, Measuring, reporting and summarizing Performance, Quantitative Principles of computer design, Pipelining: Structural, data and control hazards, linear pipeline and Non linear pipeline.

Unit 6: Instruction Level Parallelism (ILP) (8 Hrs)

ILP: Concepts and challenges, Basic Compiler Techniques for exploiting ILP, Reducing Branch costs with prediction, Overcoming Data hazards with Dynamic scheduling, Hardware based Speculation.

Unit 7: Advanced Issues in Instruction Level Parallelism (ILP) (8 Hrs)

Exploiting ILP using multiple issues and static scheduling, Exploiting ILP using dynamic scheduling, multiple issue and speculation, Advanced Techniques for instruction delivery and Speculation, The Intel Pentium 4 as example.

Unit 8: Multiprocessors and Thread –Level Parallelism (6 Hrs)

Introduction, Symmetric shared-memory architectures, Performance of symmetric shared–memory multiprocessors, Distributed shared memory and directory-based coherence.

Text Books:

1. Computer Architecture And Organization AN INTEGRAGATED APPROACH –Miles Murdocca and Vincent Heuring (WILEY).
2. John L. Hennessey and David A. Patterson: Computer Architecture, A Quantitative Approach, 4th Edition, Elsevier, 2007.
   (Chapter. 1.1 to 1.9, 2.1 to 2.10, 4.1to 4.6, Appendix A of Text Book 2)

Reference Books :

1. Computer Organisation -Hamacher Zaky(MGH)
2. Computer Architecture and Organization - John P. Hayes
3. Computer Organization and Architecture - William Stallings
4. Digital Logic and Computer Design - M. Morris Mano. Pearson Education - Prentice Hall.
5. http://virtual-labs.ac.in/labs/cse10/cla_design.html

Tutorial:

Students should do case studies based on the topics listed below:

1. What Happened to Supercomputers?
2. Patriot Missile Defense Failure Caused by Loss of Precision
3. The Java Virtual Machine
4. Associative Memory in Routers
5. The Intel Pentium 4 Memory System
6. Graphics Processing Unit
7. How a Virus Infects a Machine
8. Use the simulator to perform following experiment on computer organization and architecture.

    - Ripple Carry Adder
    - Carry-look-ahead adder
    - Booth's Multiplier
    - Arithmetic Logic Unit
    - Memory Design
    - Associative cache Design
    - Direct Mapped cache Design
    - CPU Design

SOLAPUR UNIVERSITY, SOLAPUR

T. E. (INFORMATION TECHNOLOGY)

Part – I

3. SYSTEM SOFTWARE

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

Examination Scheme
Theory: 100 Marks
Term-Work: 25 Marks
POE: 50 Marks

Course Objectives

1. To enable a student to get aquainted with different language processors.
2. To provide students necessary background to analyse, synthesize, design and develop prototypes of language processors.
3. To make students to use Language Processor Development Tools.

Course Outcomes

At the end of the course the student

1. should develop an ability to identify Language Processors.
2. should develop an ability to Analyze, Synthesize and Design language Processors.
3. should develop an ability to use Language Processor Development Tools.

SECTION-I

Unit 1: Introduction of System Software. (10 Hrs)

Introduction, language processing activities, Fundamentals of language processing, Fundamentals of language Specification, Language Processor Development Tools.

Unit2: **Compilation and Analysis. (10 Hrs)**

Aspects of compilation, Phases of Compiler, Lexical Analysis: Role of a Lexical analyzer, input buffering, specification and recognition of tokens, finite automata implications, designing a lexical analyzer generator. Syntax Analysis: 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, SLR and LALR parsers.

Unit 3: Assemblers, Macros and Macro Processors (10 Hrs)

Elements of assembly language programming, A simple assembly scheme, Pass structure of assemblers, design of a two pass assembler. Macro definition and call, Macro Expansion, Nested macro calls

SECTION II

Unit 4: Code Generation and Optimization (10 Hrs)

Code Generation: 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 Dags and the dynamic code generation algorithm.

Code Optimization: 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.

Unit 5: Linkers (10 Hrs)

Relocation and linking concepts, design of a linker, Self-relocating programs, A Linker for MS DOS, Linking for overlays

Unit 6: Loaders (10 Hrs)

Function of loader, general loader scheme, Absolute loader, Subroutine Linkages, Relocating loader, Direct linking loader, Dynamic loading, Design of an Absolute Loader, Design of direct linking loader.

Text Books:

1. System Programming and operating systems – 2nd Edition D.M. Dhamdhere (TMGH)(Unit-1, 3, 5)
2. Compilers - Principles, Techniques and Tools - A.V. Aho, R. Shethi and J.D. Ullman (Pearson Education.)(Unit-2, 4)
3. System Programming -- J. J. Donovan (Mc-Graw Hill)(Unit 6)

Reference Books:

1. System Software- An Introduction to Systems Programming- 3rd Edition- Leland L. Beck(Pearson Education)
2. Compiler Construction - Dhamdere (Mc-Millan)
3. Compiler Construction – Principles & Practice – Ken Louden (Cengage Learning)
4. Compiler Design in C – Allen I. Holub (PHI / Pearson Education)
5. Compiler Construction - Barret, Bates, Couch (Galgotia)
6. Unix Programming - Pepkin Pike.
7. Crafting a compiler with C – Charls Fischer, Richard LeBlane (Pearson Education)
8. http://nptel.iitm.ac.in for video lectures on different subjects.

Termwork:

Minimum of 8-10 practical assignments should be carried based on –

1. Symbol table generation for given input *.c file.
2. Implementation of Macro and Nested macros.
3. Design and implementation of 1 pass assemblers.
4. Design and implementation of 2 pass assemblers.
5. Design Lex specifications for the tokens – keywords, identifiers, numbers, operators, white spaces.
6. Implement any one of the code optimization techniques.
7. Implementation of Toy-code generator.
8. Simulation of linkers.
9. Simulation of loaders

SOLAPUR UNIVERSITY, SOLAPUR.

T. E. (Information Technology)

Part– I

4. Principles of Operating Systems

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

Course Objectives :

1. To acquaint students with the principles of building Operating Systems.
2. To get acquainted with the functions of the components of Operating Systems.
3. To get acquainted with different memory management schemes.

Course Outcomes :

At the end of the course

1. students will be able to exhibit the principles of operating systems through simulations.
2. students will be able to specify the functions of various components of operating systems.
3. students will be able to analyze and design prototypes to exhibit different memory management schemes.

SECTION – I

Unit 1: Introduction: (4Hrs.)

What operating systems do? , Computer system organization, Architecture, Structure, Operating system operations, Process management, Memory management, Storage management.

Unit 2: Process Concept: (4Hrs.)

Process Concept, Process Scheduling, Operation on processes, Inter-process Communication, Threads.

Unit 3: Process Scheduling: (5Hrs.)

Basic concepts, Scheduling Criteria, Scheduling algorithms, Algorithm evaluation.

Unit 4: Process Synchronization: (6Hrs.)

Background, The critical section problem, Peterson's solution, Synchronization H/W, Semaphores, Classical problems of synchronization, Monitors.

SECTION-II

Unit 5: Deadlocks: (7Hrs.)

System modes, Deadlock characterization, Methods for handling deadlocks, Deadlock, deadlock prevention, deadlock avoidance, Deadlock detection, Recovery from deadlock.

Unit 6: Memory Management: (8Hrs.)

Background, Swapping, Contiguous Allocation, Paging, Structures of page table, Segmentation.

Unit 7: Virtual Memory: **(6Hrs.)**

Background, Demand paging, Copy on write, Page replacement, Allocation of frames, Thrashing, Memory mapped files.

Unit 8: I/O System: **(5Hrs.)**

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

Text Book:

1. Operating System concepts – Silberschatz, Galvin, Gagane. (WILEY Publication).

Reference:

1. Operating Systems: Internals and Design Principles by William Stallings (PHI)
2. Operating Systems in Depth by Thomas W. Doeppner (WILEY).

Practical List:

1. Basic Unix commands.
2. Creation of a separate process using fork() system call.
3. FCFS scheduling algorithm
4. SJF (preemptive & non-preemptive)
5. Priority scheduling algorithm.
6. Round robin (RR) algorithm.
7. Peterson's solution to critical section problem.
8. Mutual Exclusion using semaphore (waits & signal).
9. Producer consumer problem (Bounded & Unbounded buffer).
10. Deadlocks with MUTEX.
11. Banker's algorithm.
12. File locking.

SOLAPUR UNIVERSITY, SOLAPUR

T. E. (INFORMATION TECHNOLOGY)

Part – I

5. DESIGN & ANALYSIS OF ALGORITHMS

Teaching Scheme	**Examination Scheme**
Theory: 3 Hrs/Week	Theory: 100 Marks
Tutorial: 1 Hr/Week	Term-Work: 25Marks

Course Objectives:

1. To acquaint students to find time complexity in terms of step count of given algorithm and express in terms of asymptotic notations.
2. To learn best, average and worst case complexity of algorithm and analyze different algorithms based on time and space complexity.
3. To study different methods of devising algorithms like divide & conquer, greedy method, dynamic programming, backtracking, branch & bound.
4. To devise an algorithm for given real world problem & find its complexity.

Course Outcomes:

After completion of course, student will be able to

1. student will be able to find step count of a given algorithm.
2. student will be able to find time & space complexity of an algorithm in terms of asymptotic notations.
3. student will be able to compare algorithms based on complexities.
4. student will be able to apply the knowledge of standard algorithm methods to solve a given problem statement.
5. student will be able to decide an algorithm devising method for a given real world problem & develop the necessary algorithm.

SECTION – I

1. **Introduction :** **(8 Hrs)**
   What is an algorithm, Algorithm Specifications, Recurrence relations, Performance Analysis

2. **Divide and Conquer :** **(9 Hrs)**
   The general method, Binary search, Finding the maximum and minimum, Merge sort, Quick sort, Selection

3. **The Greedy method :** **(9 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

4. **Dynamic Programming :** **(10 Hrs)**
   The general method, Multistage graphs, All pair shortest paths, Optimal binary search trees, 0/1 knapsack problem, Reliability design, Traveling Sales person problem, Flow shop scheduling.

5. **Backtracking** **(5 Hrs)**
   The general method, 8-queen problem, sum of subsets, Hamilton Cycle, Graph Coloring

6. **Branch and Bound** **(4 Hrs)**
   The Method, 0/1 Knapsack, Travelling Salesperson (*), Efficiency Considerations, NP-Hard and NP-Complete Problems: Basic Concepts

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. (INFORMATION TECHNOLOGY)

Part – I

6. Java Programming

Theory- 2 Hrs/Week
Practical-2 Hrs/Week

POE: 50 Marks
T/W: 25 Marks

Course Objectives :

1. To introduce the students to the features of Java programming language.
2. To be able to use the Java SDK environment to create, debug and run simple Java programs
3. To apply the Object oriented programming concepts for writing simple object oriented programs.
4. To develop skills to create GUI based application.

Course Outcomes :

At the end of the course
1. Students will be acquainted with the features of Java programming language.
2. Students will be able to apply principle of OOP concepts and explore their skills to develop a complex java program.
3. Students will be able to implement GUI based applications using Java programming.

Unit 1: Java Introduction (4 Hrs)

Overview of Java, Features of Java Language, Java Road Map, Data types and Operators, Strings, Vectors, Scanner class, Type casting, wrapper classes in Java, Define class, method, properties, Access modifiers, this keyword, Working with Constructors.

Unit 2 : Inheritance and Polymorphism: (3 Hrs)

Types of inheritance in java, Multiple inheritance using interface, Method overriding, use of super final keywords, Creating abstract classes, Interfaces & methods, Finalization and Garbage collection.

Unit 3: Exceptions and Error handling & Packages (5 Hrs)

Introduction to Exception, Dealing with exceptions, try, catch blocks, finally block, throw and throws statement, Common Exceptions, The Throwable class, User defined Exception.Understanding usage of packages. Creating user defined packages Managing classes under packages.

Unit 4: I/O Programming: (3 Hrs)

Hierarchy of classes in I/O package, Streams: Character oriented and Byte oriented, reading basic data types from keyboard. File handling in Java.

Unit 5 : Multithreading: (3 Hrs)

Java thread model, Thread creation using Thread class and Runnable Interface, Thread priorities , Thread Synchronization, Thread groups, deadlocks.

Unit 6: GUI Design in Java and Event handling: (6 Hrs)

Event Handling : Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes. The AWT class hierarchy, interface components- labels, button, canvas, scrollbars, text components, check box, check box . groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – boarder, grid, flow, card and grib bag.

Unit 7 : Applets (5Hrs)

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

Unit 8 : Java Utilities (3Hrs)

The Arrays Class, The Comaparable and Comparator Interfaces, Data structures using java: Set, Map, List, Tree, Iterators.

Text Books:-

1. The complete Reference, Java2 (5^{th} edition) – Herbert Schedt et. a (Osborn)
2. Core Java, An Integrated Apporach-Dr.R.Nageswara Rao(dreamtech)
3. Object Oriented Programming through Java- E Radhakrishna (University Press)
4. The Java Programming Language, 3^{rd} Edition – Kea Arnold, David Holmes, James Gosling, Prakash Goteti

Reference Books:-

1. The Java language specification
   (E-Book**:** http:// Java-sun.com/docs/books/jeles/downloads/langspec 3.0 pdf)
2. Programming with Java a primer – E. Balgurusamy (TMGH)
3. Java for Professionals – B.M. Harwani (SPD)

Course Instructions : Programs should be written using Notepad not by NetBeans, Eclipse or any other tool.

Practicals : Ten to Fifteen Practicals, at least one on each topic is expected.

SOLAPUR UNIVERSITY, SOLAPUR

T. E. (INFORMATION TECHNOLOGY)

Part – I

7. Self Learning Module-I(HSS)

7. 1. Economics
7. 2. Psychology
7. 3. Philosophy
7. 4. Sociology
7. 5. Humanities

The syllabus for the above subjects will be prepared centrally for all programs of Engineering.

SOLAPUR UNIVERSITY, SOLAPUR

T. E. (INFORMATION TECHNOLOGY)

Part – II

1. DATABASE ENGINEERING

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

Examination Scheme
Theory: 100 Marks
Term Work: 25 Marks
POE : 50 Marks

Course Objectives:

1. To get acquainted with the relational model of data.
2. To introduce an overview of the database-design process using the entity-relationship data model and develop query writing skills in SQL.
3. To study the different Forms of Normalization of a database.
4. To study the fundamentals of a transaction-processing system and concurrency control.

Course Outcomes:

At the end of the course,

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

SECTION-I

Unit 1: Introduction (4 Hrs)

Database-System Applications, Purpose of Database Systems, View of Data, Database Languages, Relational Databases, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and Information Retrieval, Specialty Databases, Database Users and Administrators, History of Database Systems.

Unit 2: Database Design and the E-R Model (6 Hrs)

Overview of Design Process, The Entity-Relationship Model, Constraints, Removing Redundant Attributes In Entity Sets, E-R Diagrams, Reduction to Relational Schemas, E-R Design Issues, Extended E-R Features.

Unit 3: Relational Model (10 Hrs)

Structure of Relational Databases, Database Schema, Keys, Schema Diagrams, Relational Query Languages, Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus.

Structured Query language (SQL)-Overview, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries, Modification of the Database, Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Authorization.

Unit 4: Relational Database design (8 Hrs)

Features of Good Relational Designs, Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Functional Dependency Theory, Algorithms for Decomposition, Decomposition using Multivalued Dependencies.

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, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels, Transactions 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, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Log-Based Recovery, Shadow Paging, Recovery with Concurrent Transactions, Buffer Management, Failure with Loss of Nonvolatile Storage.

Text Books :

1. Database system concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan (McGraw Hill International Edition).
2. Database system concepts by Peter Rob, Carlos Coronel (Cengage Learning).

Reference books :

1. Fundamentals of Database systems by Ramez ElMasri, S. B. Navathe (Pearson Education) .

2. Database Management Systems by Ramkrishnan Gehreke (Tata McGraw Hill).
3. Principles of Database Systems by J. D. Ullman (Galgotia Publications)
4. Advanced Database Management System by Rini Chakrabarti, Shilbhadra Dasgupta (Dreamtech Press Publication).

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 subqueries and SQL update commands.
4. a) SQL DDL and updates: write queries in SQL on schema updates.

   b) Schema Creation and constraints: write SQL queries to modify schema to create constraints.
5. Convert the created database into 1NF, 2NF, 3NF and BCNF.
6. Write a Java program for database (created in previous assignments) connectivity using JDBC.
7. Write a program to implement B+ tree index (n=3 or n=5) on the database previously created.
8. Write a program to implement dynamic hashing on the database previously created.
9. Write a program to simulate log based protocol using immediate or deferred database modification.
10. Write a program to simulate any one concurrency control protocol.
11. Given a set of functional dependencies, find canonical cover and closure of functional dependencies**.**

SOLAPUR UNIVERSITY, SOLAPUR

T. E. (INFORMATION TECHNOLOGY)

Part – II

2. Object Oriented Modeling And Design

Teaching Scheme	**Examination Scheme**
Lecture:3 Hrs/week	Theory: 100 Marks
Tutorials:1 Hr/week	Term Work**:** 25 Marks

Course objectives :

1. To Introduce students to the concepts and terms used in the object-oriented approach to systems analysis and design
2. To Highlight the importance of object-oriented analysis and design and its limitations.
3. To Show how we apply the process of object-oriented modeling and design to software development.
4. To Point out the importance and function of each model in Object Modeling Technique(OMT) & Unified Modeling Language (UML) throughout the process of object-oriented analysis and design and explain the notation of various elements in these models.

Course outcomes :

After completing this course the student must :

1. Demonstrate the knowledge and ability to show the importance of systems analysis and design in solving complex problems.
2. Show how the object-oriented approach differs from the traditional approach to systems analysis and design.
3. Explain the importance of modeling and how Object Modeling Technique (OMT) and the Unified Modeling Language (UML) represents an object-oriented system using a number of modeling views.
4. Construct various OMT & UML models for representing and analyzing real world problems.

SECTION-I

Unit 1: Introduction **(4 hrs)**

Object oriented development and themes, evidence for usefulness, modeling as a Design Technique.

Unit 2: Object Modeling (6 hrs)

Objects, classes, links and associations, generalization and inheritance, grouping constructs, aggregation, abstract classes, generalization as extension and restriction, multiple inheritance, metadata, candidate keys and inheritance.

Unit 3: Dynamic and Functional Modeling (6Hrs)

Events, states, operations, concurrency, nested state diagrams, advanced dynamic modeling concepts, relation of object and dynamic models, DFD, relation of functional to object and dynamic models.

Unit 4: Implementation of OMT (6 hrs)

Use of programming language and database system, Object oriented style, feature of object-oriented languages, Applications of OMT like object diagram compiler, Computer animation.

SECTION-II

Unit 5: Structural Modeling using UML (8 hrs)

Classes, Relationships, Common mechanisms. Diagrams, Class Diagrams, Interfaces, Types and Roles, Packages, Instances and Object Diagram.

Unit 6: Behavioral Modeling using UML (8hrs)

Interactions, Use cases, Use case diagram, Interaction Diagrams and Activity diagrams, Events and signals, State Machines, Processes and Threads, Time and space, State chart diagrams.

Unit 7: Architectural Modeling using UML (7 hrs)

Components, Deployment, Collaboration, Patterns and Frame works, Component diagrams and Deployment Diagrams.

Text Books:

1. Object oriented Modeling and Design: Rambaugh, Premerlani, Eddy, Lorenson (PHI)
2. The Unified Modeling Language User Guide: Grady Booch, Jeams Rambaugh, Ivar Jacotson (Addison Wesley)

Reference Books :

1. Practical Object Oriented Design with UML – Mark Priestley.
2. UML-In a Nut Shell – Sinon Alhair.

Term work :

Student should design the following systems using :

1. Object Modeling Technique(OMT)
2. Unified Modeling Language(UML)
   a) Hospital Management System
   b) College Automation System
   c) Hotel Management System
   d) Banking System
   e) Library Information System
   f) Railway Reservation System
   g) Water Management System
   h) Supermarket Information System

SOLAPUR UNIVERSITY, SOLAPUR
T. E. (INFORMATION TECHNOLOGY)

Part – II

3. Unix Operating System Concepts

Teaching Scheme	**Examination Scheme**
Theory: 4 Hrs/Week	Theory – 100 Marks
Practical: 2 Hr/Week	Term Work - 25 Marks
	Oral Examination: 25 Marks

Course objectives:

Students will obtain:

1. To get acquainted with basic Unix commands at Shell and Kernel level.
2. To introduce advanced concepts and design issues of Unix operating systems.
3. To get in depth knowledge and explore internals of Unix OS.

Course outcomes:

At the end of the course

1. a student will get acquainted with the basic Unix commands at Shell and Kernel level.
2. a student will able to analyse, design and develop Applications in Unix environments.
3. a student will be acquainted with the internals of Unix OS.

SECTION-I

Unit1: Introduction (6 Hrs.)

General Overview of the Unix System: History, System Structure, User Perspective, Operating System Services, Assumption about Hardware, Introduction to the KERNEL: Architecture of UNIX OS, Introduction to system concepts, Kernel Data Structure, System Administration.

Unit2: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.

Unit3: Internal Representation of Files (8 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.

Unit4: System calls for the file System (10 Hrs.)

Open, Read, write, File and Record Locking, Adjusting the position of FILE I/O-LSEEK, Close, File Creation, Creation of Special File, Change Directory and Change Root, Change Owner and Change Mode, Stat and Fstat, Pipes, Dup, Mounting and Unmounting file systems, Link, Unlink, File System Abstractions, File system maintenance.

SECTION-II

Unit5: The Structure of process **(8 Hrs.)**

Process states and transitions, layout of system memory, The context of a process, saving context of a process, manipulation of the process address space.

Unit6: Process Control **(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.

Unit7: Memory management policies **(8 Hrs.)**

Swapping, Demand paging, a hybrid system with demand paging and swapping.

Unit8: The I/O Subsystem **(6 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 :

1. Unix concepts and administration by Sumitabha Das (TMGH).
2. Advanced Programming in the Unix Environment by W.Richard Stevens.

Practical List:

1. Copying the content of a file.
2. fork() system call.
3. Buffers on hash queues.
4. Scenarios for retrieval of a buffer.
5. pipe() system call.
6. setuid() system call.
7. profil() system call.
8. System call for time and alarm.
9. Basic Shell programming
    a. Finding PATH for the specified programs.
    b. A script to convert the specified filenames to lower case.
    c. The scripts convert GIF files to PNG files via the intermediate PPM format.
    d. Count the number of time each different word occurs in the files given as arguments.
10. Display the type of a command line file.
11. malloc() system call.

SOLAPUR UNIVERSITY, SOLAPUR
T. E. (INFORMATION TECHNOLOGY)
Part – II

4. Software Engineering

Teaching Scheme	Examination Scheme
Lecture : 3 Hrs/week	Theory **:** 100 Marks
Tutorials: 1 Hr/week	Term Work **:** 25 Marks

Course objectives :

The Course should enable the student

1. To be acquainted with the software life cycle models.
2. To get acquainted with modeling and modeling languages.
3. To design and develop correct and robust software products.
4. To analyze business requirements pertaining to software development.

Course outcomes :

At the end of the course

1. a student will be able to select proper software life cycle model for design and development.
2. a student will be able to prepare prototype models using standard modeling languages.
3. a student will be able to focus on the fundamentals of developing a Software Project.
4. a student will be able to get Software Requirement Specification from client, analyze, design and estimate the cost of development of a Software Project.

SECTION-I

Unit 1: Introduction to Software Engineering (9Hrs.)

Software Development Life Cycle, Life cycle models : Water fall model, Iterative development model , Spiral model, Rational unified Process model, Prototype model, Time Boxing model, Agile process model. Project management process, software configuration management process.

Unit 2: Software Requirement Analysis and Specification (10Hrs)

Value of Good SRS, Requirement Process, Requirements specification, Functional Specification with Use Cases, Other Approaches for Analysis : Data Flow Diagram, Entity Relationship Diagram. Validation, Software Architecture : Role of Software Architecture, Architecture Views , Component & Connector View, Architecture Style for Component & Connector view, Documenting Architecture Design.

Unit 3: Design (6 Hrs.)

Design Concepts: Coupling, Cohesion, Open Closed Principle,
Function-Oriented Design, Object Oriented Design, Detailed Design, Verification, Metrics.

SECTION-II

Unit 4: Planning a Software Project (8Hrs.)

Effort estimation, Project Schedule and Staffing ,Quality planning: Quality Concepts, Qualitative quality management planning. CMM project management process, Risk Management Planning, Project Monitoring Plan, Detailed Scheduling.

Unit 5: Agile Project Management (6Hrs.)

Introduction to APM, Implementation, Iterative Project Management Life Cycle, Adaptive Project Management Life Cycle, Adaptive & Integrating the APM toolkit

Unit 6: Testing (6Hrs.)

Testing Concepts , Tesing Process , Black-Box Testing, White-Box Testing, Object Oriented Software testing methods, Functional testing, Unit testing, System testing, User satisfaction testing.

Text Books:

1. Pankaj Jalote's Software Engineering, A Precise Approach(Wiley Precise Textbook,WILEY INDIA)
2. An Integrated Approach to Software Engineering- 3rdedition: Pankaj Jalote (Narosa Publishers)

Reference Books :

3. Effective Project Management Traditional,Agile,Extreme ,Robert K. Wysocki WILEY INDIA,6th edition.
4. Ian Sommerville,software engineering, pearson education Asia, 6th edition
5. Software Engineering Fundamentals –Ali Behforooz and Frederick j. Hudson (Oxford University Press)

Term work:

In Tutorial Session, Students of Different Batches should be assigned Different Case Studies to Design & Implement.

SOLAPUR UNIVERSITY, SOLAPUR

T. E. (INFORMATION TECHNOLOGY)

Part – II

5. Mobile Application Development

Teaching Scheme
Lectures: 04 hrs/week

Examination Scheme
Theory: 100 Marks
Term Work : 25 Marks

Course Objectives:

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

Course Outcomes:

1. Students will be able to familiarize with mobile apps development aspects.
2. To design & develop mobile apps, using Android as a development platform.
3. To perform testing, signing, packaging and distribution of mobile apps.

Section – I

Unit I: Android Operating System (8)

Introduction, History, Features and Characteristics, Ecosystem, Hardware Requirements, Development Model, Android Concepts, Overall Architecture.

Unit-II: Getting started with Mobility (8)

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

Unit -III: User Interface Design. (06)

App user interface designing – mobile UI resources (Layout, UI elements, Draw-able, Menu), Activity- states and life cycle, interaction amongst activities.

Unit -IV: Mobile Application Functionality. (06)

App functionality beyond user interface - Threads, Async task, Services – states and life cycle, Notifications, Broadcast receivers, Telephony and SMS APIs

Section – II

Unit-V: Native data handling **(6)**

On-device file I/O, shared preferences, mobile databases such as SQLite, and enterprise data access (via Internet/Intranet)

Unit -VI: Sprucing up mobile apps **(10)**

Graphics and animation – custom views, canvas, animation APIs, multimedia – audio/video playback and record, location awareness, and native hardware access (sensors such as accelerometer and gyroscope)

Unit -VII: Testing mobile apps **(6)**

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

Unit VIII: Taking apps to Market **(6)**

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

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).

Term Work : The Term Work should consist of completing the following assignments.

List of Assignments:

Students should implement and learn to use the android application development and testing tools to accomplish the following assignments during regular course schedule.

1. Understand the app idea and design user interface/wireframes of mobile app
2. Set up the mobile app development environment
3. Using emulator to deploy and run mobile apps
4. Develop and debug mobile app components – User interface, services, notifications, broadcast receivers, data components.
5. Testing mobile app - unit testing, black box testing and test automation.

SOLAPUR UNIVERSITY, SOLAPUR
T. E. (INFORMATION TECHNOLOGY)
Part – II

6. Advanced Java

Teaching Scheme
Theory- 2 Hrs/Week
Practical-2 2Hrs/Week

Examination Scheme
POE: 50 Marks
T/W: 25 Marks

Course Objectives :

1. To introduce the advanced features of Java programming language (Sockets, RMI, JDBC, Java Beans).
2. To develop the ability to create networking & distributed applications using Java programming.
3. To develop the ability to create web & database based applications.

Course Outcomes :

At the end of the course
1. Students will be acquainted with the advanced features of Java programming language.
2. Students will be able to implement database based applications using Java programming.
3. Students will be in a position to develop networking, distributed and web applications.

SECTION-I

Unit 1: Swings (4 Hrs)

Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing-JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

Unit 2: Network programming with Java (3 Hrs)

Networking fundamentals,Client server programming: InetAddress, URLS, Sockets, Datagram Socket,One way & Two way Communication, Retrieving a file at server.

Unit 3: Remote Method Invocation (RMI) (4 Hrs)

Introduction,RMI Architecture,The Remote Interface,The Remote Object,Writing the Server,The RMI Compiler,Writing the Client,Remote Method Arguments and Return Values,Dynamic Loading of Stub Classes

Unit 4: Java Database Connectivity (JDBC) (4 Hrs)

Introduction, Relational Databases, JDBC Architecture, java.sql & javax.sql package, JDBC vs ODBC, Various types of JDBC driver(Type-1,Type-2,Type-3, Type-4),Structured Query Language, Transactions, Meta Data.

SECTION-II

Unit 5: Servlets **(5 Hrs)**

Definition, Basic Servlet Application Programming Interface (API), Servlet Architecture, Servlet Lifecycle, Creating Servlet Application file and Deploying application, Connecting to the database using Servlets, Cookies & Sessions.

Unit 6: JSP **(5 Hrs)**

Introduction to JSP, JSP vs. Servlet, JSP Architecture, Life cycle of JSP, JSP Elements, JSP Documents, Action Elements, Connecting to the database using JSP, Cookies & Sessions.

Unit 7: Enterprise Java Beans **(4 Hrs)**

Introduction, Types of Enterprise Bean, Writing Enterprise Beans, Beginning with Enterprise Java Beans, Working with session beans.

Unit 8: Struts **(3 Hrs)**

Struts Framework Basic, Understanding Stuts, MVC Architecture, Setting up Struts, Struts Flow Control

Text Books:-

1. Java Server Programming for Professionals - Ivan Bayross, Sharanam Shah, Cynthia Bayross and Vaishali Shah, 2nd Edition, Shroff Publishers Distributors Pvt. Ltd.
2. Core Java, Volume II – Advanced Features - Cay S. Horstmann, Gary Cornel, 8th Edition
3. Core Servlets and Java Server Pages, Volume 2, Advance Technology, 2nd Edition – Marty Hall, Larary Brown, Yaakov Chaikin
4. Java 7 Programming Black Book-Dream Tech Press.

Reference Books:-

1. Head First Sevlets and JSP – Bryan Bosham, Kathy Sierra, Bert Bates, O'Reily Publication
2. The complete Reference, Java2 (5th edition) – Herbert Schedt et. a (Osborn)

Course Instructions:

Using Eclipse or Net Beans practical should be conducted. Servlets and JSP programs should be deployed on Tomcat web server not by Net Beans or Eclipse.

Tools to be Used:

- Java SDK 6.0
- Eclipse
- Net Beans Platform
- Tomcat / Web Logic / Websphere

Practicals:

Ten to Fifteen practical at least one on each topic is expected.

SOLAPUR UNIVERSITY, SOLAPUR.

T. E. (INFORMATION TECHNOLOGY)
Part– II

7. Seminar

Practicals : 2Hrs/week **Term work : 25 Marks**

Course objectives:

1. To study, analyze & prepare a topic for presentation on existing or new technology.
2. To exhibit effective communication.
3. To work in teams having brainstorming session for group discussion.

Course outcomes :

At the end of the course

1. Student will get acquainted with an existing or a new technology.
2. Student will exhibit good communication & presentation skills.
3. Student will be able to discuss, brainstorm & work in teams.

.

The groups of students of strength 4 should be formed by the end of T.E.-Sem-I. The project areas for the group should be finalized by the end of 1st month of T.E.-II. Seminar should consist of a presentation of about 30-40 minutes by every individual student. The seminar should be based on topics in the area in which the students have carried on the literature survey and will work for their selected project (whose title is finalized in TE Part – II) in the final year. A report on the seminar should be submitted to the department. Assessment should be jointly done by panel of teachers consisting of respective guide and other teachers from the department.

Solapur University, Solapur

T.E. (INFORMATION TECHNOLOGY)

Part-II

8. Self Learning Module-II: HSS/Technical

8.1. Network Setup and Management

Examination Scheme
Theory: 50 Marks

Course Objectives:

1) To study and understand the working of network devices.
2) To get acquainted with working of various active components in the computer networking environment.

Course Outcome:

At the end of the course

1) Students will be to identify various networking devices.
2) Students will be able to setup and manage the computer network.

Unit 1: Hubs and Switches

Hubs, Switches, Switch Types, Planning a Chassis-Based Switch Installation.

Unit 2: VLANs

Connecting VLANs, Configuring VLANs, CatOS, IOS Using VLAN Database, IOS Using Global Commands, Nexus and NX-OS

Unit 3: Router and Routing

Routing Tables, Router Types, The IP Routing Table, Virtual Routing and Forwarding.

Unit 4: Routing Protocols

Communication Between Routers, Metrics and Protocol Types, Administrative Distance, Routing Protocols: RIP, RIPv2, EIGRP, OSPF, BGP.

Unit 5: Firewall

The Best Practices, DMZ, Alternate Designs

Unit 6: Wireless

Wireless Standards, Security, Configuration of WAP, MAC Address filtering, Troubleshooting.

Unit 7: Designing Network

Documentation: Requirements Documents, Port Layout Spreadsheets, IP and VLAN Spreadsheets, Bay Face Layouts, Power and Cooling Requirements, Tips for Network Diagrams. Naming

Conventions for Devices, Network Designs: Corporate Networks, Ecommerce Websites, Modern Virtual Server Environments, Networks.

Unit 8: IPv6

Addressing, Subnet Masks, Address Types, Subnetting, NAT, Simple Router Configuration.

Text Books:

4. Network Warrior: Gary A. Donahue , OREILLY Publication.
5. TCP/IP Protocol Suite : Behroz A. Forozen (Third Edition)

Reference Book :

1. Andrew S. Tanenbaum, Computer Networks, Prentice Hall.

SOLAPUR UNIVERSITY, SOLAPUR.

T. E. (Information Technology)
Part – II
8. Self Learning Module-II (HSS/Technical)
8.2. Tools for Computer Architecture

Examination Scheme
Theory: 50 marks

Course Objectives:

1) To expose the students to the various key aspects of Computer Organization & Architecture.
2) To introduce students to design real world problems and conduct experiment on computer organization & Architecture using simulators.

Course Outcome:

At the end of the course

1. Students can put into practice gate level design to CPU design.
2. Student will able to identify, formulate and solve advanced computer architecture problems.

Unit 1: The Instruction Set Architecture

Introduction and study of a RISC Computer: ARC Processor, Input and Output in assembly language, Data path and Control, the assembly process, linking and loading.

Unit 2: Study of ARC Tool (SPARC Processor)

ARC tools, The ARC Assembler, Loading, Assembling, and Examining a File, Saving Files, Loading Files into the Simulator, Measuring Program Performance, The Time Model , configuration Editor, Memory/IO Parameters, Time Model's Statistics Window, The Cache Simulator View.

List of Experiments

1. Using ARC Simulator Perform the following experiments
2. Write a subroutine to perform a swap operation on 32 bit operands
3. Write an interface for an appropriate input and output devices
4. Write a program that flashes the screen every time when the user's position changes (like finger on touch screen).

Perform FPGA based prototyping of experiments with support of a virtual environment. Download simulator from virtual lab (Ministry of Human Resource and Development (MHRD) http://virtual-labs.ac.in/labs/cse10/rca_design.html)

- Ripple Carry Adder
- Carry-look-ahead adder
- Registers and Counters
- Wallace Tree Adder
- Combinational Multipliers
- Booth's Multiplier
- Arithmetic Logic Unit
- Memory Design
- Associative cache Design
- Direct Mapped cache Design
- CPU Design

Text Books:

1. Computer architecture and organization: an integrated approach Miles J. Murdocca, Vincent P. Heuring. (Ch 4,5,6) (App B)

References Books:

1. Computer Organization and Architecture - William Stallings
2. Computer System Architecture - M. Morris Mano
3. Computer Architecture and Organization - John P. Hayes

Web Sites:

- http://en.wikipedia.org/wiki/Wallace_tree
- http://www.ecs.umass.edu/ece/koren/architecture/
- NPTEL (e-learning courses from IITs and IISC)

Solapur University, Solapur
T.E. (Information Technology)
Part-II
8. Self Learning Module-II: HSS/Technical
8.3. Compiler Development Tools

Examination Scheme
Theory: 50 Marks

Course Objectives:

1. Establish an understanding of the concepts and tools used for the development of compilers .
2. Provide experience in the development of a completes/ working compiler for a substantial language.

Course Outcomes:

At the end of the course

1. a student will be able to describe stages of compilation.
2. a student will be able to code lexical analyzer using a tool.
3. a student will be able to code parser using a tool.

Unit 1: Lex and Yacc

The Simplest Lex Program, Recognizing Words with Lex, Grammars, The Parts of Speech Lexer, Running Lex and Yacc, Lex vs. Hand-written Lexers.

Unit 2: Using Lex

Regular Expressions, a Word Counting Program, Parsing a Command Line, a C Source Code Analyzer.

Unit 3: Using Yacc

Grammars, Shift/Reduce Parsing, a Yacc Parser, The Lexer, Arithmetic, Expressions and Ambiguity, Variables and Typed Tokens, Symbol Tables, Functions and Reserved Words, Building Parsers with Make.

Unit 4: A Reference for Lex Specifications

Structure of a Lex Specification, BEGIN, ECHO, Input from Strings, Internal Tables (%N Declarations), Line Numbers and yylineno, Multiple Lexers in One Program, output(), Portability of Lex Lexers, Regular Expression Syntax, REJECT, Returning Values from yylex(), unput(), yyinput(), yyoutput(), yyunput(), yyleng, yyless(), yylex(), yymore(), yytext, yywrap().

Unit 5: A Reference for Yacc Grammars

Structure of a Yacc Grammar, Actions, Ambiguity and Conflicts, Bugs in Yacc, End Marker, Error Token and Error Recovery, %ident Declaration, Inherited Attributes ($0), Lexical Feedback, Literal Block, Literal Tokens, Portability of Yacc Parsers, Precedence, Associativity, and Operator Declarations, Recursive Rules, Rules, Special Characters, Start Declaration, Symbol Values, Tokens, %type Declaration, %union Declaration, Variant and Multiple Grammars, y.output Files, Yacc Library, YYABORT, YYACCEPT, YYBACKUP, yyclearin, yydebug and YYDEBUG, YYDEBUG, yyerrok, YYERROR, yyerror(), yyparse(), YYRECOVERING().

Unit 6: Yacc Ambiguities and Conflicts

The Pointer Model and Conflicts, Common Examples of Conflicts, IF—THEN—ELSE, How to Fix the Conflict?, IF—THEN—ELSE (Shift/Reduce).

Unit 7: Error Reporting and Recovery

Error Reporting, Error Recovery, Compiler Error Recovery.

Text Book:

Lex & Yacc- John R. Levine, Tony Mason, Doug Brown, 2nd/updated edition (October 1992) O'Reilly & Associates**.**

Reference book:

Lex & Yacc -Levine, Mason, Brown, 2nd edition, O'Reilly & Associates, Inc.