SOLAPUR UNIVERSITY, SOLAPUR

FACULTY OF ENGINEERING & TECHNOLOGY

INFORMATION TECHNOLOGY

Syllabus for

T.E. (Information Technology) w.e.f. Academic Year 2018-19

Choice Based Credit System

SOLAPUR UNIVERSITY, SOLAPUR

FACULTY OF ENGINEERING & TECHNOLOGY

Information Technology

Programme Educational Objectives and Outcomes

A. Program Educational Objectives

1. To build a strong foundation in mathematics, science & technology in students required to prepare them for Graduate studies and research.
2. To prepare students to apply knowledge of core & application domain, to analyze & design complex engineering problems using latest technologies & tools.
3. To develop effective communication, presentation skills and management principles in students and enable them to apply these in their work as a member or a leader in a team for managing projects.
4. To promote awareness for life-long learning, environment, sustainability, health & safety, economics etc. in students and to introduce them to professional ethics to build a good social personality.

B. Program Outcomes

Engineering Graduate 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

The program must enable students to attain, by the time of graduation:

1. Get acquainted and apply fundamentals of mathematics, science and core information technologies and become aware of the processes that support the delivery and management of information systems within a specific application environment.
2. An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems appending them with the ability to use and apply latest technologies and tools in the core information technologies with following best practices and standards
3. An ability to effectively integrate IT-based solutions into the user environment.
4. Live and work as a contributing, well rounded member of society.

SOLAPUR UNIVERSITY, SOLAPUR
Faculty of Engineering & Technology
Third Year Information Technology

Choice Based Credit System Syllabus Structure of T.E. Information Technology W.E.F. 2018-2019

Semester I

Course Code	***Theory Course Name***	***Hrs./week***			***Credits***	***Examination Scheme***			
		L	***T***	***P***		***ISE***	***ESE***	***ICA***	***Total***
IT311	Principles of Operating System	3	--	--	3	30	70	--	100
IT312	System Software	3	--	--	3	30	70	--	100
IT313	Design and Analysis of Algorithms	3	1	--	4	30	70	25	125
IT314	Database Engineering	3	--	--	3	30	70	--	100
IT315	Computer Organization and Architecture	3	1	--	4	30	70	25	125
IT316	Java Programming	3	--	--	3	--	--	--	--
SLH31	Self Learning Module-I	--	--	--	2	--	50	--	50
	Sub Total	18	2	_	22	150	400	50	600
Course Code	***Laboratory Course Name***								
							ESE		
							POE	***OE***	
IT311	Principles of Operating System	_	_	2	1	_	50	--	25	75
IT312	System Software	_	_	2	1	_	--	_	25	25
IT314	Database Engineering	--	--	2	1	--	50	--	25	75
IT316	Java Programming	_	_	4	2	_	50	_	25	75
	Sub Total	--	--	10	5	_	150		100	250
	Grand Total	**18**	**2**	**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)*

SOLAPUR UNIVERSITY, SOLAPUR
Faculty of Engineering & Technology
Third Year Information Technology

Choice Based Credit System Structure of T.E. Information Technology W.E.F. 2018-2019

Semester II

Course Code	***Theory Course Name***	***Hrs./week***			***Credits***	***Examination Scheme***			
		L	***T***	***P***		***ISE***	***ESE***	***ICA***	***Total***
IT321	Unix Operating System	4	--	--	4	30	70	--	100
IT322	Software Engineering	3	1	--	4	30	70	25	125
IT323	Object Oriented Modeling and Design	3	1	--	4	30	70	25	125
IT324	Artificial Intelligence	3	--	--	3	30	70	--	100
IT325	Mobile Application Development	3	--	--	3	30	70	--	100
IT326	Python Programming	2	--	--	2	--	--	--	--
IT327	Self Learning Module-II	--	--	--	2	--	50	--	50
	Sub Total	18	2	--	22	150	400	50	600
Course Code	***Laboratory Course Name***								
							ESE		
							POE	***OE***	
IT321	Unix Operating System	--	--	2	1	--	--	25	25	50
IT324	Artificial Intelligence	--	--	2	1	--	--	--	25	25
IT325	Mobile Application Development	--	--	2	1	--	50	_	25	75
IT326	Python Programming	--	--	2	1	--	50	_	25	75
IT328	Seminar	--	--	2	1		--	--	25	25
	Sub Total		--	10	5	--	125		125	250
	Grand Total	**18**	**2**	**10**	**27**	**150**	**525**		**175**	**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)*

1. Batch size for the practical /tutorial shall be of 15 students. On forming the batches, if the strength of remaining student exceeds 7, then a new batch shall be formed.

2. 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

3. 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

4. Curriculum for Humanities and Social Sciences Self Learning Modules is common for all under graduate programmes of faculty of Engineering and Technology

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. Minimum four assignments for Self Learning Modules at T.E. Part I and T.E. Part II shall be submitted by the students which shall be evaluated by a Module Coordinator assigned by institute / department

7. Project groups for B.E. (I.T.) Part I and Part II formed at T.E. (I.T.) Part II for seminar shall not be of more than five students.

8. ICA assessment shall be a continuous process based on student's performance in – class tests, assignments, homework, subject seminars, quizzes, laboratory books and their interaction and attendance for theory and lab sessions as applicable

Solapur University, Solapur
T.E.(Information Technology) Semester -I
IT311 PRINCIPLES OF OPERATING SYSTEMS

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

Introduction: Principles of Operating Systems is introductory course in operating systems. This course teaches the fundamentals of operating system. The core of the course contains process concept, process scheduling and synchronization, concurrent programming, threads, inter process communication, deadlocks and memory management concepts.

Course Prerequisite: Student should know the concept of Digital techniques, Microprocessor and Data Structures in C.

Course Objectives:

1. To introduce the basic concepts and functions of operating system.
2. To make student aware of process management and thread management.
3. To implement different process scheduling algorithms.
4. To implement the concept of concurrency control, Memory Management, I/O & File management.

Course Outcomes:

At the end of the course student will

1. get acquainted with the basic concepts and functions of operating systems.
2. be able to design and develop Applications process management and thread management in Unix environments.
3. be to develop process scheduling algorithms of Unix OS.
4. be able to write memory Management Techniques, develop the concept of concurrency control and develop applications on file Management techniques in Unix OS.

SECTION-I

Unit 1: Introduction to Operating System **04 Hrs**

What an Operating System can do?, Components of Operating System, Operating System Structure, Operating System Operations, Objectives and Functions, Evolution of Operating Systems – Simple Batch System, Multi-programmed Batch System, Time Sharing System, Personal Computer System, Parallel System, Real Time System and System Structures

Unit 2: Process Concept **05 Hrs**

Process Concept, Process States, Process Control Block, Operations on Process, Inter-process Communication (IPC) – Shared memory systems and Message passing systems, Examples of IPC Systems, Threads – Processes and Threads, Concept of Multithreading, Multithreading Models, Thread Libraries – Win32 Threads and P-threads

Unit 3: Process Scheduling **06 Hrs**

Basic Concepts – CPU-I/O Burst Cycle, CPU Scheduler and Dispatcher, Scheduling Criteria, Scheduling Algorithms - First-Come, First-Served Scheduling, Shortest-Job-First Scheduling, Shortest-Remaining-Time-First Scheduling, Priority Scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling and Multilevel Feedback Queue Scheduling, Algorithm Evaluation - Deterministic Modeling, Queuing Models and Simulations

Unit 4: Process Synchronization **04 Hrs**

Background, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Monitors.

SECTION-II

Unit 5: Deadlocks **07 Hrs**

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

Unit 6: Memory Management: **05 Hrs**

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

Unit 7: Virtual Memory: **06 Hrs**

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

Unit 8: Input / Output and File Management **08 Hrs**

I/O Management and Disk Scheduling: I/O Devices, Organization of the I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling (FIFO, SSTF, SCAN, C-SCAN, LOOK, C-LOOK), Disk Cache. File Management: Overview, File Organization and Access, File Directories, File Sharing, Record Blocking, Secondary Storage Management.

Text Book:

1. Operating System concepts – Silberschatz, Galvin, Gagane. (WILEY Publication).
2. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, ISBN-10: 0-13380591-3, ISBN-13: 978-0-13-380591-8, 8th Edition.

Reference:

1. Harvey M. Deitel, Operating Systems, Prentice Hall, ISBN-10: 0131828274, ISBN-13: 978-0131828278
2. Thomas W. Doeppner, Operating System in depth: Design and Programming, WILEY, ISBN: 978-0471-68723-8

Suggested List of Lab Assignments of Principles of Operating System

1. Basic Unix commands.
2. Creation of a separate process using fork () system call.
3. FCFS scheduling algorithm
4. SJF (pre-emptive & 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 sharing & Record Blocking

Solapur University, Solapur
T. E. (Information Technology) Semester– I
IT312. SYSTEM SOFTWARE

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

Introduction: This course introduces different language processors. It deals with methods used to analyze, synthesize, design and develop prototypes of language processors. It introduces some tools to be used to develop language processors.

Course Prerequisite: Student shall have undergone a course on Discrete Mathematical Structures and Theory of Computation.

Course Objectives

1. To enable a student to get acquainted with different language processors.
2. To provide students necessary background to analyze, 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 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. 06 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 08 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 **06Hrs**

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

Unit 6:Loaders **05 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.

Internal Continuous Assessment (ICA):

Minimum of 8 practical assignments should be carried based on –

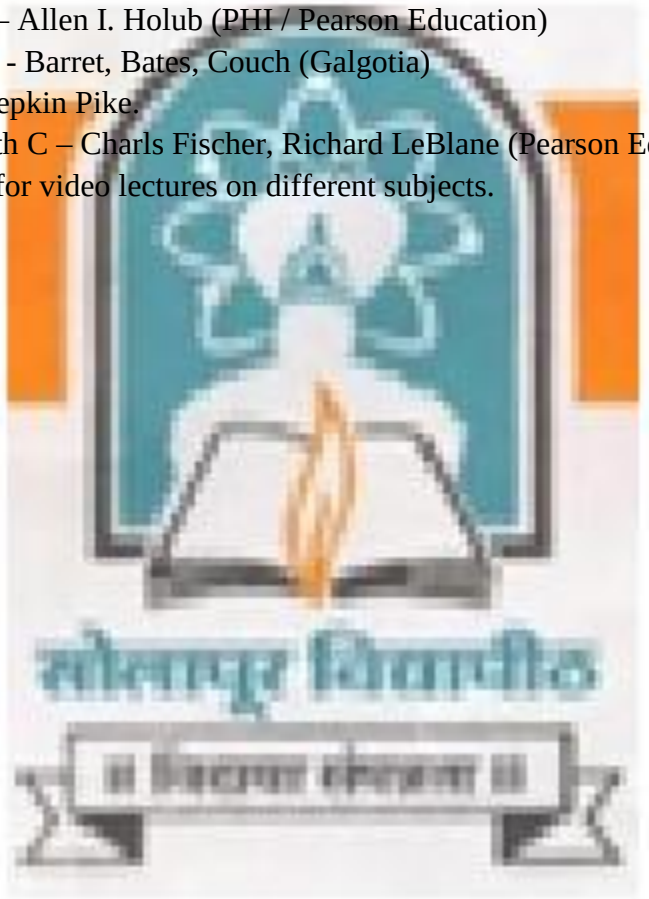
1. Symbol table generation for given input *.c file.
2. Implementation of Macro and Nested macros.
3. Design and implementation of 1 pass assemblers.
4. Design and implementation of 2 pass assemblers.
5. Design Lex specifications for the tokens – keywords, identifiers, numbers, operators, white spaces.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

Text Books:

5. System Programming and operating systems – 2nd Edition D.M. Dhamdhere (TMGH)(Unit-1, 3, 5)
6. Compilers - Principles, Techniques and Tools - A.V. Aho, R. Shethi and J.D. Ullman (Pearson Education.)(Unit-2, 4)
7. 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.



Solapur University, Solapur
T.E. (Information Technology) Semester-I
IT 313 DESIGN OF ALGORITHM AND ANALYSIS

Teaching Scheme	**Examination Scheme**
Lectures : 3 Hours/week, 3 Credits	**ESE : 70 Marks**
Tutorial : 1 Hour/week, 1 Credit	**ISE : 30 Marks**
	ICA : 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. To acquaint students to find time complexity in terms of step count of given algorithm and express in terms of asymptotic notations
2. To get acquainted with terms best, average and worst case complexity of the algorithm and analyze different algorithms based on time and space complexity
3. To acquire the knowledge of various methods of devising algorithm like divide and conquer, greedy method, dynamic programming, backtracking, branch and bound etc.
4. To design an algorithm for given real-world problem and find its complexity
5. To acquire knowledge of P, NP, NP-complete and NP-Hard Problem and apply them to differentiate between tractable and intractable problems.

Course Outcomes:

1. Students will be able to find step count of a given algorithm.
2. Students will be able to find time and space complexity of an algorithm in terms of asymptotic notations.
3. Students will be able to compare algorithm based on complexities
4. Students will be able to apply knowledge of standard algorithm methods to solve a given real-world problem statements
5. Students will be able to differentiate the P, NP and NP-Complete problems.

Section I

Unit 1:Introduction **08 Hrs**

What is an algorithm? Algorithm Specifications, Recurrence relations, Performance Analysis

Unit 2: Divide and Conquer **09 Hrs**

The general method, Binary Search, Finding the maximum and minimum, Merge sort, Quick sort, Selection sort.

Unit 3:The Greedy Method **09 Hrs**

The general method, knapsack problem, Job sequencing with deadlines, minimum cost spanning trees-Prim's and Krushkal's Algorithms, Optimal storage on tapes, Optimal merge patterns., Single source shortest paths.

Section II

Unit 4: Dynamic Programming **04 Hrs**

The general method, multistage graphs, All pair shortest paths,graphs, All pair shotest paths, Optimal binary search tress, 0/1 knapsack problem, Reliability design, Traveling Sales person problem, Flow shop scheduling.

Unit 5:Backtracking **05 Hrs**

The general method, 8-queen problem, sum of subsets, Hamilton Cycle, Graph Coloring

Unit 6:Branch and Bound **04 Hrs**

The Method, 0/1 Knapsack, Travelling Salesperson (*), Efficiency Considerations, NP-Hard and NP-Complete Problems: Basic Concepts

---

- **Term Work:**
  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. (Information Technology) Semester-I
IT 314 DATABASE ENGINEERING

Teaching Scheme
Lectures: 3Hours/week, 3 Credits
Practical: 2 Hour/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 Perquisites:

Math foundations: elementary set theory, concepts of relations and functions

Data structures: trees, B-trees, linear data structures, dictionaries, graphs.

Algorithms: Basic algorithm design methods and techniques for algorithm complexity analysis
Programming languages: a general purpose programming language

Course Objectives:

1. To get acquainted with the relational model of data.
2. To introduce 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 a student will be able to,

1. Demonstrate the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.
2. Design E-R diagrams to represent simple database for any real time application and formulate

SQL queries on it.

3. Analyze a database design and improve the design by normalization.
4. Demonstrate knowledge of ACID properties of a transaction and several techniques of concurrency control.

SECTION-I

Unit 1: Introduction 04 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 06 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 Multi-valued Dependencies.

SECTION – II

Unit 5: Indexing and Hashing **07 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 **07 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 **06 Hrs**

Lock-Based Protocols, Deadlock Handling, Multiple Granularity, Timestamp-Based Protocols, Validation-Based Protocols.

Unit 8: Recovery System **08 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 (6^{th} edition,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).
5. SQL The Complete Reference(Third Edition, McGraw Hill International Edition)

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) Semester - I
IT315 COMPUTER ORGANIZATION & ARCHITECTURE

Teaching Scheme	Examination Scheme
Lectures: 4 Hours/Week, 3 Credits	**ESE: 70 Marks**
Tutorial: 1 Hours/Week, 1 Credit	**ISE: 30 Marks**
	ICA: 25 Marks

Introduction: Computer Organization and Architecture (COA) course provide students with an understanding of the design of fundamental blocks used for building a computer system and interfacing techniques of these blocks to achieve different configurations of an "entire computer system". It introduces detailed understanding of various processor micro architectural designs, which include pipeline design, and multi-core processor design.

Course Prerequisite:

Student shall have undergone a course on Digital Logic Design and Operating system.

Course Objectives:

1. To enable the student to get acquainted with the history and development of modern computers and its basic principles
2. To analyze the performance of computer systems and design prototypes of computers.
3. To deal with issues relating to modern processors like SPARC and use cache and pipelining to improve efficiency.

Course Outcome:

At the end of the course a student should be able to

1. Identify the major components of a computer including CPU, memory, I/O and storage, assembly language programming, uses for cache memory.
2. Carry on cost performance analysis and design prototypes for computers.
3. Acquainted with assembly language of SPARC Processor and use of cache memory.

SECTION-I

Unit 1: Introduction **04 Hrs**

A Brief History of Computing, The Von Neumann Model, Generations of Computers, The System Bus Model, Levels of Machines: Upward Compatibility, The Levels of computer, A Typical Computer System.

Unit 2: Data Representation and Arithmetic **12 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 and Memory **09 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, The Memory Hierarchy, Cache Memory.

Unit: 4 Fundamentals of Computer Design **05 Hrs**

Introduction, Classes of Computers, Defining Computer Architecture, Trends in Technology, Power in Integrated Circuits and cost, Dependability, Measuring, Reporting and Summarizing Performance, Quantitative Principles of computer design.

SECTION-II

UNIT: 5 Fundamentals of Pipeline: **08 Hrs**

Introduction to Pipelining, The Major Hurdle of Pipelining: Pipeline Hazards, linear pipeline and Non linear pipeline.

.

UNIT: 6 Instructions –Level Parallelism -1: **08 Hrs**

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

UNIT: 7 Instructions –Level Parallelism -2: **08 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.

UNIT: 8 Multiprocessors and Thread –Level Parallelism: **06 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)

References :

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 or http://192.168.1.7:8080/vlab/

Tutorial:

The Objective is to expose the students to the various key aspects of Computer Organization & Architecture.
Students should do case studies based on the topics below:

1. **Case Study**: What Happened to Supercomputers?
2. **Case Study:** Patriot Missile Defense Failure Caused by Loss of Precision
3. **Case Study**: The Java Virtual Machine
4. **Case Study**: Associative Memory in Routers
5. **Case Study**: The Intel Pentium 4 Memory System
6. **Case Study**: Graphics Processing Unit
7. **Case Study**: 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) Semester - I
IT 316 JAVA PROGRAMMING

Teaching Scheme	**Examination Scheme**
Lectures: 3Hours/Week, 3 Credits	**ICA: 25 Marks**
Practical: 4Hours/Week, 2 Credits	**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 Prerequisite: Students must be familiar with basic programming languages like C or Python

Course Objectives:

1. To get acquainted with 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 **05 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 **06 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, Interfaces, Using the Keyword "this" , Object Life time & Garbage Collection, Recursion in Java, Type Casting.

Unit 3: Exceptions, Error Handling and Basic IO **05 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 **06 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 **06 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 **06 Hrs**

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

Unit 7: JDBC and RMI **06 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 **05 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.

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

Term Work:

- 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.

Solapur University, Solapur
T.E. (Information Technology) Semester-II
IT321 UNIX OPERATING SYSTEM

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

Introduction: The UNIX operating system is a common platform in industry and is used by many research organizations. It includes fundamental concepts of operating systems, emphasizing a hands-on introduction to the UNIX file system, task management, common system utilities, the UNIX programming environment. Students gain experience with system installation and administration.

Course Prerequisites: C Programming, Fundamental of Data Structures, Microprocessor

Course objectives:

Students will be able

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.
4. To demonstrate the functioning of system calls in 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.
4. A student will develop the system program for the functioning of system call in Unix OS.

SECTION-I

Unit1: General Overview of the Unix System **06 Hrs**

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 **06 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 **08 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, lose, 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 **08 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 **08 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 **08 Hrs**

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

Unit8: The I/O Subsystem **06 Hrs**

Driver interfaces, Disk drives, Terminal drivers, Streams.

Text Books:

1. The design of Unix Operating System - Maurice J. Bach (PHI)
2. Unix Manuals.
3. Operating System Concepts" by Avi Silberschatz and Peter Galvin

Reference Books:

1. Unix concepts and administration by Sumitabha Das (TMGH).
2. Advanced Programming in the Unix Environment by W. Richard Stevens.
3. UNIX Systems Programming by Kay Robbins and Steve Robbins, Prentice Hall, ISBN-13: 978-0134424071, ISBN-10: 0134424077, 2nd Edition.

Suggested List of Lab Assignments of Unix Operating System Concepts

1) Installation of Unix Operating System.
2) Write the Unix commands & Perform their Operations.
3) Write a Program to copy the contents of One file to another file.
4) Write a program to demonstrate the various scenarios the kernel follows to allocate a buffer for a disk block.
5) Write a program to implement Fork () System call.
6) Write a program that prints the Owner, file type, access permissions, access times of files supplied as parameters. If a file is a directory, the program should read the directory and print the above information for all files in the directory.
7) Write a program to implement for Unnamed pipe system call.
8) Write a program to implement for Named pipe system call.
9) Write a program to implement System call for time and alarm.
10) Write a shell program to count the number of time each different word occurs in the Files

Solapur University, Solapur
T. E. (Information Technology) Semester-II
IT322 SOFTWARE ENGINEERING

Teaching Scheme	**Examination Scheme**
Lecture : 3 Hours/week, 3 Credits	**ESE :70 Marks**
Tutorials: 1 Hour/week, 1Credit	**ISE: 30 Marks**
	ICA: 25 marks

Introduction:

Software Engineering focuses on the principles and knowledge of software engineering. It covers the approaches taken in developing software projects, including requirements analysis and specification, design, coding, testing, maintenance and thorough documentation as illustrated by examples. This course will prepare students for working in teams to build quality software, and it provides the necessary hands-on practice for those who wish to enhance their knowledge base.

Course Prerequisite:

Student shall have undergone a course on Object Oriented Programming through C++. An understanding of Object Oriented Analysis and Design, Programming Skills.

Course objectives:

The Course should enable the student

1. To be acquainted with the software development life cycle models.
2. To analyze business requirements pertaining to software development.
3. To design and develop correct and robust software products.
4. To widen managerial skills for planning, implementing and risk management of software Project.
5. To analyze and apply the basic types of testing.

Course outcomes:

At the end of the course student will be able to

1. Make use of suitable models through analysis of requirements and arrive at an appropriates of m software design
2. Prepare SRS document for a project
3. Apply software design and development techniques
4. Analyze and Apply project management techniques for a case study
5. Generate test cases for various applications.

SECTION-I

Unit 1: Introduction to Software Engineering **11 Hrs**

Introduction, The Problem Domain, Software Engineering Challenges and Approach, Software Process, Characteristics of Software Process, Software Development Process Models: Water fall model, Iterative development model, Spiral model, Rational unified Process model, Prototype model, Time Boxing model, Agile process model.

Unit 2: Software Requirement Analysis & Specification **06 Hrs**

Value of Good SRS, Requirement Process, Requirements specification, Functional Specification with Use Cases, Other Approaches for Analysis: Data Flow Diagram, Entity Relationship Diagram.

Unit 3: Software Architecture and Design **08 Hrs**

Software Architecture: Role of Software Architecture, Architecture Views, Component & Connector View, Architecture Style for Component & Connector view, Documenting Architecture Design.
Design Concepts: Coupling, Cohesion, Open Closed Principle, Function-Oriented Design, Object Oriented Design, Detailed Design, Verification, Metrics.

SECTION-II

Unit 4: Testing **06 Hrs**

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

Unit 5: Project Planning and Management **08 Hrs**

Project management process, The Inspection Process, software configuration management process, 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 6: Agile Project Management **06Hrs**

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

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)
3. Effective Project Management Traditional,Agile,Extreme ,Robert K. Wysocki WILEY INDIA,6th edition.

Reference Books :

1. Ian Sommerville,software engineering, pearson education Asia, 6th edition
2.Software Engineering Fundamentals –Ali Behforooz and Frederick j. Hudson (Oxford University Press)

Term work:

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

Case Study:

The purpose of the case study is to illustrate the entire development life cycle and focus on methodology, deliverables, steps, techniques used for project development.

Students develop a Web site based on problem statement assigned by faculty during the tutorial session.

Case Study 1: Library Book Circulation System

Case Study 2: Online Railway Reservation System

Solapur University, Solapur
T. E. (Information Technology) Semester– II
IT323: OBJECT ORIENTED MODELING AND DESIGN

Teaching Scheme	**Examination Scheme**
Lectures: 3 Hours/Week, 3 Credits	**ESE: 70 Marks**
Tutorial: 1 Hour/Week, 1 Credit	**ISE: 30 Marks**
	ICA: 25Marks

Introduction: This course presents Object Oriented approaches to software development based on modeling objects from the real world and then using the model to build a language independent design organized around those objects. These techniques promote better understanding of requirements, cleaner designs and more maintainable systems.

Course Prerequisite: Student shall have undergone an introductory course on Object oriented Design and Programming.

Course Objectives:

1. To Introduce students to the concepts and terms used in the object-oriented approaches.
2. To enable students to analyze real world problems and create an Object model to represent static view of the system.
3. To enable students to analyze real world problems and create a Dynamic model to represent behavioral view of the system.
4. To enable students to analyze real world problems and create a Functional model to represent process view of the system.
5. To enable students to use the above three models and create a system design and Object design.
6. To train students to use UML as a tool to create models in the form of diagrams.

Course Outcomes:

After completing this course the student will be able to :

1. Demonstrate the knowledge about concepts used in Object Oriented approaches.
2. Create the Object Model, Dynamic model, Functional model and use these models to create the system design and Object Design required for software development.
3. Draw respective UML Diagrams representing designs to be used for software development.

SECTION-I

Unit 1: Introduction to Object Oriented approach and Object Modeling. 10 Hrs

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

Unit2: Dynamic and Functional Modeling 06 Hrs

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 3: Implementation of OMT **06 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 4: Structural Modeling using UML **08 Hrs**

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

Unit 5: Behavioral Modeling using UML **08 Hrs**

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 6: Architectural Modeling using UML **07 Hrs**

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

Internal Continuous Assessment (ICA):

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

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

Solapur University, Solapur
T. E. (Information Technology) Semester– II
IT324: ARTIFICIAL INTELLIGENCE

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

Introduction: This course presents a basic introduction to the techniques used in developing Artificial Intelligent systems. It is a walkthrough to problem spaces and search algorithms, Knowledge representation, reasoning, logic programming and applications of Artificial Intelligence

Course Prerequisite: Student shall have some exposure to algorithms and programming.

Course Objectives:

1. To introduce students to identify artificial intelligent systems.
2. To enable students to analyze search algorithms for solving problems that is candidates for AI techniques.
3. To enable students to represent real world problems using various Knowledge representation methods
4. To enable students to use techniques to reason under uncertainty and having statistical data.
5. To enable students to use AI techniques to develop applications like games, plans and expert systems.

Course Outcomes:

After completing this course the student will be able to:

1. Demonstrate the knowledge about Artificial Intelligent systems.
2. Solve real world problems using appropriate searching algorithms
3. To represent problems using an appropriate knowledge representation method.
4. Reason under uncertainty.
5. To develop applications that is candidates for AI.

SECTION-I

Unit 1: Introduction to Artificial Intelligence **06 Hrs**

Definition of AI, The AI problems, the understanding Assumption, AI Technique, the level of the model, criteria for success.

Unit2: Problem spaces and Search **10 Hrs**

Definition of a problem as a state space search, Problem characteristics, Issues in the design of search programs

State Space Search: Depth First Search, Breadth First Search, DFID. Heuristic Search: Best First Search, Hill Climbing, Beam Search, Tabu Search.

Randomized Search: Simulated Annealing, Genetic Algorithms, Ant Colony Optimization.

Finding Optimal Paths: Branch and Bound, A*, IDA*, Divide and Conquer approaches, Beam Stack

Search.
Problem Decomposition: Goal Trees, AO*, Rule Based Systems, Rete Net.

Unit 3:Knowledge Representation **06 Hrs**
Representation and Mappings, Approaches to Knowledge representation, Issues in knowledge representation, The frame problem. Representing simple facts in predicate logic, Representing instances and Isa Relationships, computable functions & Predicates, Resolution, Natural deduction, Procedural versus declarative knowledge, logic programming, Forward versus backward reasoning, matching, control knowledge.

SECTION-II

Unit 4: Symbolic Reasoning under uncertainty and Statistical Reasoning : 08Hrs
Introduction to non monotonic reasoning, logic for non monotonic reasoning, implementation issues, probability and Baye's Theorem in certainty factor and Rule – Based systems, Bayesian Networks, Demster Shafer Theory, Fuzzy logic.

Unit 5: Logic Programming with Prolog **08 Hrs**
Backward Chaining, Logic Programming, Prolog, Search in Prolog, Controlling Search, The Cut Operator in Prolog.

Unit 6: Applications of AI **07 Hrs**
Planning and Constraint Satisfaction: Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Graph plan, Constraint Propagation.
Natural Language Understanding: Natural Language Semantics, CD Theory, English to CD Theory
Game Playing: Min-max Algorithm, Alpha-Beta Algorithm, SSS*.
Expert System: Introduction, Rule Based System Architectures, Knowledge Acquisition and validation, Knowledge system building tools.

Internal Continuous Assessment (ICA):
Student should implement the following:

1. Breadth first search, Depth first search and Best first search algorithms.
2. Implementation of a simple Genetic Algorithm.
3. A* and AO* algorithms.
4. Developing an application using PROLOG.
5. Min-Max and Alpha Beta Algorithm

Text Books:

1. Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.
2.Artificial Intelligence - Elaine Rich, Kevin Knight, Shivashankar B. Nair (Third Edition) , Tata McGraw Hill, 2009.

Reference Books:

1. Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, Morgan Kaufmann, 2011.
2. John Haugeland, Artificial Intelligence: The Very Idea, A Bradford Book, The MIT Press, 1985.
3. Pamela McCorduck, Machines Who Think: A Personal Inquiry into the History and

Prospects of Artificial Intelligence, A K Peters/CRC Press; 2 edition, 2004.
4. Zbigniew Michalewicz and David B. Fogel. How to Solve It: Modern Heuristics. Springer; 2nd edition, 2004.
5. Judea Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, 1984.
6. Elaine Rich and Kevin Knight. Artificial Intelligence, Tata McGraw Hill, 1991.
7. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.
8. Eugene Charniak, Drew McDermott. Introduction to Artificial Intelligence, Addison-Wesley, 1985.
9. Patrick Henry Winston. Artificial Intelligence, Addison-Wesley, 1992.

Solapur University, Solapur
T. E.(Information Technology) Semester-II
IT325 MOBILE APPLICATION DEVELOPMENT

Teaching Scheme	**Examination Scheme**
Lectures : 03 hours/week, 3 Credits	**ESE: 70 Marks**
Practical : 02 hours/week, 1 Credit	**ISE: 30 Marks**
	ICA: 25 Marks
	POE: 50 Marks

Introduction: Mobile application development course will build your skills in creating native mobile apps for Android platform. This course includes Android application development with basic User Interface design, basic building blocks, data handling in Android App, Graphics, Animation and Multimedia in Android, Testing mobile apps and how to taken app to the market.

Course Prerequisite: Java or object-oriented programming experience.

Course Objectives:

1. To learn the basics of Android platform and get to understand the application lifecycle.
2. To develop mobile applications using modern mobile development tools for android.
3. To develop mobile apps using basic components such as Activity, Intent, ContentProvider, Services and Broadcast receivers.
4. To implement mobile application functionality beyond user interface using Threads, Async Task concepts.
5. To design and implement mobile apps based native data handling such as on-device file I/O, shared preferences, and mobile databases like SQLite.
6. To explain the concept of graphics and animations, multimedia, native hardware access, testing, signing, packaging and distribution of mobile apps.
7. To implement a mini-project in a team of 3-5 members using mobile app development concepts.

Course Outcomes:

1. Student is able to explain the features, characteristics, hardware requirement, architecture of Android Operating System and different approaches to develop an app.
2. Student is able to setup mobile app development environment with an emulator using Android Studio.
3. Student is able to develop mobile apps using UI resources and basic components such as Activity, Intent, ContentProvider, Services and Broadcast receivers.
4. Student is able to implement mobile application functionality beyond user interface using Threads, Async Task concepts.
5. Student is able to design and implement mobile apps based native data handling such as on-device file I/O, shared preferences, and mobile databases like SQLite.
6. Student is able to explain the concept of graphics and animations, multimedia, native hardware access, testing, signing, packaging and distribution of mobile apps.
7. Student is able to implement a mini-project in a team of 3-5 members using mobile app development concepts.

Section – I

Unit 1: Mobile Platform – Android **04 Hrs**

Introduction, Mobility Panorama, Mobile Platforms, App Development Approaches, Android Overview and History, Android Features, Android Stack, Android Architecture

Unit-2: Getting Started with Android **04 Hrs**

SDK Overview, Setting Up Android App Development Environment using Android Studio, Creating First Hello World App – creating first project, setting up an Emulator – Running first app on emulator, Behind the Scenes - Android app execution flow, Android App project structure, Logical Components of Android App

Unit -3: Basic Android User Interface Design **08 Hrs**

UI resources - Layout resources, String Resources, Image Resources, UI Elements (Button, EditText, CheckBox, RadioButton, RadioGroup, ListView, ImageView, Dialogs – AlertDialog, ProgressDialog, TimePickerDialog, and DatePickerDialog), Event-Handling Paradigm (Event Source, Event Object, Event Listener, Event handler), Event-handling implementation with event objects – Button, CheckBox, RadioGroup, ListView, ImageView etc.)

Unit -4: Android App Building Blocks **08 Hrs**

Logical Components of an Android App, Activity – Activities, Activity Life-cycle &callback methods, Interaction among Activities - Intents, Navigation between Activities, Exchanging Data, Threads, AsyncTask, Service – States and Life-cycle methods, Initiating a Service, IntentService, BoundService, Notifications, Broadcast Receivers, Telephony and SMS

Section – II

Unit-5: Data Handling in Android App **06 Hrs**

File System and On-device file I/O, Shared Preferences, Mobile Database - Introducing SQLite, SQLiteOpenHelper and creating a database, Opening and closing a database, Working with cursors Inserts, updates, and deletes and enterprise data access (via Internet/Intranet)

Unit -6: Graphics, Animation and Multimedia in Android **06 Hrs**

Android Graphics – Supporting Multiple Screens, Drawables, Custom View and Canvas, Android Animation – Drawable Animation, View Animation, Property Animation, Multimedia – audio/video playback and record, location awareness, and native hardware access (sensors such as accelerometer and gyroscope)

Unit -7: Testing mobile apps **05 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 **04 Hrs**

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).

ICA (Internal Continuous Assessment) evaluation should consist of following assignments. Students should design and implement the following Android Application using Android Studio (System Requirement: 8 GB RAM)

List of Assignments:

1) Design and implement "Hello World!!!" First Android application.
2) Design and implement an android application using Android UI elements (Button, EditText, CheckBox, RadioButton, RadioGroup, ListView, ImageView)
3) Design and implement an android application using String Resources and Intent concept.
4) Design and implement an android application using Dialogs (AlertDialog, ProgressDialog, TimePickerDialog, and DatePickerDialog)
5) Design and implement an android application using Thread and Async Task concept.
6) Design and implement an android application using Service component (Service, IntentService, BoundService).
7) Design and implement an android application using Notification concept.
8) Design and implement an android application to read and write contents to Flat File on
   (i) internal memory of mobile device and
   (ii) external memory (SD card) of mobile device.
9) Design and implement an android application using "Shared Preferences" concept to change theme of your application.
10) Design and implement an android application using SQLite to insert, delete, update, and display Student information into database.
11) Write JUnit test cases for Android application.
12) Mini-project – Design and develop an android application for real-time problems.

Solapur University, Solapur
T.E. (Information Technology) Semester-II
IT 326 PYTHON PROGRAMMING

Teaching Scheme	**Examination Scheme**
Lectures: 2 Hours/week, 2 Credits	**ESE : 50 Marks**
Practical: 2 Hour/week, 1 Credit	**ICA : 25 Marks**

Introduction: Python is a popular, general-purpose, multi-paradigm, open-source, scripting language. It is designed to emphasize code readability – has a clean syntax with high level data types. It is suited for interactive work and quick prototyping, while being powerful enough to write large applications. This course introduces the python language which has simple syntax, powerful set of libraries and robust debugger and profiler.

Course Prerequisite:

Student should have knowledge of basic programming.

Course Objectives:

1. Introduce to the core components of programming using the Python programming language.
2. Introduce library packages to write desktop applications using python
3. To impart basic understanding of python programming with GUI and Database programming

Course Outcomes:

1. Use fundamental library packages available in python,
2. Design python application using procedure oriented and object oriented approach.
3. Design and Develop GUI and database application in python,

Section I

Unit 1:Introduction **04 Hrs**

Introducing the Python Interpreter, Program Execution, Execution Model Variations, The Interactive Prompt, System Command Lines.

Unit 2: Introduction to procedural programming in Python **06 Hrs**

Data types, Collection data types, Control structures and functions, Custom Functions, Powerful Lamda function in python,String, String handling, command line programming, time and dates, List, Dictionary and TupleSet

Unit 3: Modules and packages **07 Hrs**

Modules, Packages, Commonly used Python Modules, Math, Random, Time, Regular Expression, JSON handling, File and directory handling, Create, read, write delete, and rename files, Traverse directories, PyPI: Python Package Index, pypi.python.org/pypi, Using pip to install python packages from PyPI , NumPy, SciPy,

Section II

Unit 4: Object oriented programming and Database Programming **05 Hrs**

Attributes and methods, Inheritance and polymorphism, Unit testing and profiling, databases, Executing Queries, SQL databases

Unit 5: Basic GUI Programming and Multithreading **04 Hrs**

Introduction to Tkinter programming, Tkinter widgets , Thread , Starting a thread , Threading module , Synchronizing threads.

Unit 6: Testing, Debugging and exceptions **04 Hrs**

Testing output, Unit tests in Python, Exception Handling, Handling Multiple exceptions, creating custom exceptions, Debugging programs

---

- **Term Work:**
  Term work consists of minimum eight to ten experiments based upon above curriculum.

Text Book:

1. Programming in Python 3, Second Edition, Mark Summerfield

References:

1. Python Cookbook, Third Edition, David Beazley and Brian K. Jones, Shroff Publishers & Distributors Pvt. Ltd., ISBN : 978-93-5110-140-6
2. Learning Python FIFTH EDITION Mark Lutz
3. Programming Python (English) 4Th Edition Mark Lutz
4. Testing Python, David Sale, Wiley India (P) Ltd., ISBN : 978-81-265-5277-1

Solapur University, Solapur
T.E. (Information Technology) Semester-II
IT 327 Self Learning
327.1 Network Setup and Management

Teaching Scheme	**Examination Scheme**
Self learning	**ESE : 50 Marks**

Introduction :

Course Prerequisite: Student should know the basics of network devices, cables and networking tools.

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:

1. Network Warrior: Gary A. Donahue , OREILLY Publication.
2. 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) Semester-II
IT 327 Self Learning
327. 2 Tools for Computer Architecture

Teaching Scheme	**Examination Scheme**
Self learning	**ESE : 50 Marks**

Intorduction:

Tools for Computer Architecture course provide students with an understanding of using tools to design the fundamental blocks used for building a computer system and interfacing techniques of these blocks to achieve different configurations of an entire computer system.

Course Prerequisite: Student shall have undergone a course on Digital Logic Design and Operating system.

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) Semester-II
IT 327 Self Learning
327. 3 Compiler Development Tools

Teaching Scheme	**Examination Scheme**
Self learning	**ESE : 50 Marks**

Introduction:
The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language or machine code). The tools permit us to develop the six phases of a compiler.

Course Prerequisite: System software and microprocessor

Course Objectives:
1. To get acquainted with the concepts and tools used for the development of compilers.
2. Provide experience in the development of the phases of 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.

Solapur University, Solapur
T.E. (Information Technology) Semester-II
IT 327 Self Learning
327. 4 Data Science

Teaching Scheme
Self learning:

Examination Scheme
ESE : 50 Marks

Intorduction:

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.

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. (Information Technology) Semester-II
IT 327 Self Learning
327. 5 User Interface Technologies

Teaching Scheme	**Examination Scheme**
Self learning	**ESE : 50 Marks**

Introduction: The main objective of UI Technology is to to make the user's interaction as very simple and most efficient. UI helps us to decrease the gap between requirements and implementation over structured systems associated with the programming language.

Course Prerequisites:

1. **Basic HTML**: Structure and HTML Tags, Images, List, Tables, Anchors and Form Elements
2. **CSS** : Inline, Internal and External Style sheet, Borders, Backgrounds, text and margin properties

---

Course objectives:

Students will be able

1. To understand the concepts and architecture of the World Wide Web.
2. To understand and practice Markup Language.
3. To understand and practice Embedded Dynamic Scripting on Client-side Internet Programming.
4. To understand and practice Web Development Techniques on client-side.

Course Outcomes:

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

1. Acquire knowledge about functionalities of World Wide Web
2. Explore markup languages features and create interactive web pages using them
3. Learn and design Client-side validation using scripting languages
4. Acquire knowledge about Open source JavaScript libraries
5. Able to design Front-end web page.

Unit I: Introduction to WWW – 6 Hrs

Introduction to Computer networks - Internet Standards – Introduction to WWW – WWW Architecture – SMTP – POP3 – File Transfer Protocol - Overview of HTTP, HTTP request – response — Generation of dynamic web pages.

Unit II: UI Design: 12 Hrs

HTML5: What is HTML5 - Features of HTML5 – Semantic Tags – New Input Elements and tags - Media tags (audio and video tags) – Designing Graphics using Canvas API - Drag and Drop features – Geo location API - Web storage (Session and local storage).

CSS3: What is CSS3 –Features of CSS3 – Implementation of border radius, box shadow, image border, custom web font, backgrounds - Advanced text effects(shadow) - 2D and 3D Transformations - Transitions to elements - Animations to text and elements

Unit III: Responsive Web Design (RWD): 9 hrs

Responsive Design: What is RWD – Introduction to RWD Techniques – Fluid Layout, Fluid Images and Media queries- Introduction to RWD Frame work

Twitter Bootstrap – Bootstrap Background and Features - Getting Started with Bootstrap - Demystifying Grids – Off Canvas - Bootstrap Components - JS Plugins - Customization

Unit IV: Introduction to JavaScript : 12hrs

Introduction - Core features - Data types and Variables - Operators, Expressions and Statements - Functions & Scope - Objects - Array, Date and Math related Objects - Document Object Model - Event Handling –Browser Object Model - Windows and Documents - Form handling and validations.

Object-Oriented Techniques in JavaScript - Classes – Constructors and Prototyping (Sub classes and Super classes) – JSON –Introduction to AJAX.

Unit V: Introduction to jQuery : 6hrs

Introduction – jQuery Selectors – jQuery HTML - Animations – Effects – Event Handling – DOM – jQuery DOM Traversing, DOM Manipulation – jQuery AJAX

Courseware & reference books:

The courseware reference books and web links can be used:

1. Harvey & Paul Deitel& Associates, Harvey Deitel and Abbey Deitel, “Internet and World Wide Web - How To Program”, Fifth Edition, Pearson Education, 2011.
2. Achyut S Godbole and Atul Kahate, “Web Technologies”, Second Edition, Tata McGraw Hill, 2012.
3. Thomas A Powell, Fritz Schneider, “JavaScript: The Complete Reference”, Third Edition, Tata McGraw Hill, 2013.
4. David Flanagan, “JavaScript: The Definitive Guide, Sixth Edition”, O'Reilly Media, 2011
5. Bear Bibeault and Yehuda Katz, “jQuery in Action”, January 2008
6. Web link for Responsive Web Design - https://bradfrost.github.io/this-is-responsive/
7. Ebook link for JavaScript - https://github.com/jasonzhuang/tech_books/tree/master/js

Solapur University, Solapur
T.E. (Information Technology) Semester-II
IT 327 Self Learning
327. 6 AGILE PROJECT MANAGEMENT

Teaching Scheme	**Examination Scheme**
Self learning	**ESE: 50 Marks**

Introduction: Agile Project Management course exposes students to Agile Software Development methodology. This course includes agile fundamentals, scrum framework, agile testing, agile software design & programming and industry trends.

Course Prerequisite: Awareness of basics of software engineering concepts and waterfall methodology and exposure to any object oriented programming language such as Java, C#.

Course Objectives:

1. To study the background and driving forces for taking an Agile approach to software development.
2. To study the business value of adopting Agile approaches.
3. To study the Agile development practices.
4. To make development with unit tests using Test Driven Development.
5. To apply design principles and refactoring to achieve Agility.
6. To study automated build tools, version control and continuous integration.
7. To perform testing activities within an Agile project.

Course Outcomes:

At the end of this course, student is able to:

1. Explain the background and driving forces for taking an Agile approach to software development.
2. Explain the business value of adopting Agile approaches and the Agile development practices.
3. Drive development with unit tests using Test Driven Development.
4. Apply design principles and refactoring to achieve Agility.
5. Explain automated build tools, version control and continuous integration.
6. Perform testing activities within an Agile project.

Unit I: Fundamentals of Agile:
The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project management, Design and development practices in Agile projects, Test Driven Development, Continuous Integration, Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools

Unit II: Agile Scrum Framework:
Introduction to Scrum, Project phases, Agile Estimation, Planning game, Product backlog, Sprint backlog, Iteration planning, User story definition, Characteristics and content of user stories, Acceptance tests and Verifying stories, Project velocity, Burn down chart, Sprint planning and retrospective, Daily scrum, Scrum roles – Product Owner, Scrum Master, Scrum Team, Scrum case study, Tools for Agile project management

Unit III: Agile Testing:
The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester

Unit IV: Agile Software Design and Development:
Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control (SVN tool)

Unit V: Industry Trends:
Market scenario and adoption of Agile, Agile ALM, Roles in an Agile project, Agile applicability, Agile in Distributed teams, Business benefits, Challenges in Agile, Risks and Mitigation, Agile rapid development technologies

Reference Books & E-Books:

1. Agile Software Development, Principles, Patterns and Practices, By Robert C. Martin, Publisher: Prentice Hall, Published: 25 Oct 2002
2. Agile Software Development with Scrum, By Ken Schawber, Mike Beedle, Publisher: Pearson, Published: 21 Mar 2008
3. Agile Testing: A Practical Guide for Testers and Agile Teams, By Lisa Crispin, Janet Gregory, Publisher: Addison Wesley Published: 30 Dec 2008

Following websites provides articles/ freely downloadable eBook on Agile Software Development:
www.it-ebooks.info/tag/agile
http://martinfowler.com/agile.html

Solapur University, Solapur
T.E. (Information Technology) Semester-II
328. Seminar

Teaching Scheme	**Examination Scheme**
Practical : 2 Hours/Week, 1 Credit	**ICA :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