Course Code: 22CS204/22CS251

**COMPUTER ORGANIZATION**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation |  | : | 40 Marks |
| Tutorial | : | - | Semester End Examination |  | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration |  | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | The purpose of the course is to introduce principles of computer organization and basic architectural concepts |
| 2 : | It begins with the basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations |
| 3 : | Topics include computer arithmetic, instruction set design, microprogrammed control Unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors |

**Unit I**

**Digital Computers**: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design, and Computer Architecture.

**Register Transfer Language and Micro operations**: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro-operations, shift micro-operations, Arithmetic logic shift Unit.

**Basic Computer Organization and Design**: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input–Output and Interrupt.

**Unit II**

**Microprogrammed Control**: Control memory, Address sequencing, microprogram example, design of control Unit.

**Central Processing Unit**: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, and Program Control

**Unit III**

**Data Representation**: Data types, Complements, Fixed Point Representation, Floating Point Representation.

**Computer Arithmetic**: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating–point Arithmetic operations. Decimal Arithmetic Unit, Decimal Arithmetic operations.

**Unit IV**

**Input-Output Organization**: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access.

**Memory Organization:** Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

**Unit V**

**Reduced Instruction Set Computer**: CISC Characteristics, RISC Characteristics.

**Pipeline and Vector Processing:** Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

**Multi Processors**: Characteristics of Multiprocessors, Interconnection Structures, Inter-processor arbitration, Inter-processor communication and synchronization, Cache Coherence.

**Course Outcomes**: At the end of the course, the student should be able to

|  |  |  |
| --- | --- | --- |
| CO 1 : | Understand the basics of instruction sets and their impact on processor design |  |
| CO 2 : | Demonstrate an understanding of the design of the functional Units of a digital computer system |  |
| CO 3 : | Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory. |  |
| CO 4 : | Design a pipeline for consistent execution of instructions with minimum hazards. |  |
| CO 5 : | Recognize and manipulate representations of numbers stored in digital computers. |  |

**Textbooks:**

1. Computer System Architecture, M. Morris Mano, Third Edition, Pearson/PHI, 2014.

**References:**

1. Computer Organization, Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5th edition, Mc Graw Hill, 2011.
2. Computer Organization and Architecture, William Stallings, 6th edition, Pearson/PHI, 2002.
3. Structured Computer Organization, Andrew S. Tanenbaum, 4th edition, PHI/Pearson, 2013.

Course Code: 22CB201

**JAVA BASED OBJECT ORIENTED PROGRAMMING**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation |  | : | 40 Marks |
| Tutorial | : | - | Semester End Examination |  | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration |  | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To understand object-oriented concepts and problem-solving techniques |
| 2 : | To obtain knowledge about the principles of inheritance and polymorphism |
| 3 : | To implement the concept of packages, interfaces, exception handling and concurrency mechanism |
| 4 : | To understand the Annotations, Java Database Connectivity Architecture |

**Unit I**

**Introduction to Object Oriented Programming:** Introduction to object-oriented programming concepts (Encapsulation, abstraction, Inheritance, polymorphism, etc.), Difference between JDK, JRE, and JVM, Introduction to Classes, Objects, Class Diagram and Object Diagram, Data Types, Variables, Type Conversion and Casting, Operators, Control Statements: Decision Making, Looping and Branching, Argument Passing Mechanism, Command Line Arguments, Constructors and types, Arrays and types.

**Unit II**

**Inheritance, Packages, Interfaces and Other Topics:** Understanding static, Use of “this” keyword, Garbage Collection and finalize () Method, Introduction and Working with Lambda Variables.

**Inheritance:** Basics, Using super, creating a multi level hierarchy, Constructors in Inheritance, final with Inheritance,

**Polymorphism:** Introduction and Types, Overloading and Overriding, Dynamic method Dispatch

**Packages:** Packages, Access protection, Importing Packages, Interfaces, Default Interfaces, Default interface methods, Use static methods in an Interface,

**String Handling:** Introduction and Types, Operations, Immutable String, Method of String class, String Buffer and String Builder class

**Unit III**

**Exception Handling:** Exception handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch clauses, Nested try statements, throw, throws, finally, Java Built-in Exceptions, Creating your own exception subclasses, propagating exceptions Chained Exceptions, Assertions and Localizations Concepts and its working.

**I/O Streams:** The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and Writing Files, Random access file operations, The Console class, Serialization, Enumerations, Type Wrappers.

**Unit IV**

**Multithreaded Programming, Annotations:** The java Thread Model, The main thread, Creating Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Interthread Communication, Suspending, resuming and stopping threads, Obtaining a thread state, Using Multithreading.

**Annotations:** Introduction, Custom Annotations and Applying Annotations.

**Unit V**

**Generics and Collections Framework (Java.Util):** Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Hashtable, Properties, Stack, Vector, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner, Working with generic class, interfaces.

**Database programming:** Types of Drivers, JDBC Architecture, Basic steps in developing JDBC applications, JDBC classes and Interfaces, Creating a new database and table with JDBC, Transaction Management, Stored Procedures.

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |  |
| --- | --- | --- |
| CO 1 : | Design and implement object-oriented concepts like encapsulation, abstraction and data hiding using programming constructs offered by java language. |  |
| CO 2 : | Realize the power of inheritance, interfaces, and packages. |  |
| CO 3 : | Understand and demonstrate the concepts of exception handling and java io streams. |  |
| CO 4 : | Demonstrate knowledge and understanding of multi-threading, annotations, in Java. |  |
| CO 5 : | Solve problems using java collection framework and Build the applications using JDBC API to access the database to perform CRUD Operations. |  |

**Textbooks:**

1. Java The complete reference, Herbert Schildt, 9th Edition McGraw Hill Education (India) Pvt. Ltd., 2014
2. Java How to Program, Paul Dietel, Harvey Dietel, 10th Edition, Pearson Education,2014

**References:**

1. Core Java Volume -1 Fundamentals, Cay S. Horstmann, 9th Edition, Pearson Education, 2012
2. Java Programming for core and advanced learners, Sagayaraj, Dennis, Karthik and Gajalakshmi, University Press, 2018
3. Introduction to Java programming, Y. Daniel Liang, 12th Edition, Pearson Education, 2013
4. Object Oriented Programming through Java, P. Radha Krishna, University Press, 2016

**Online resources:**

1. https://www.w3schools.com/java/java\_oop.asp

2. http://peterindia.net/JavaFiles.html

Course Code: 22CS203/22CS253

**DATABASE MANAGEMENT SYSTEMS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation |  | : | 40 Marks |
| Tutorial | : | - | Semester End Examination |  | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration |  | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To introduce the role of database management systems in an organization. |
| 2 : | To represent real-world scenarios using E-R diagrams. |
| 3 : | To modal the database using relations avoiding redundancies. |
| 4 : | To learn transaction management and concurrency protocols to ensure data consistency. |
| 5 : | To understand the database file organization system and database recovery techniques. |

**Unit I - Introduction to DBMS**

History of DBMS, Concepts, and overview of DBMS, Data models - ER model, Relational model, Levels of Abstraction in DBMS, Database Languages, Architecture of DBMS, Data Base Users and Administrators.

**ER-Model**

Database design and ER model, ER modeling Constructs, Additional features of ER Model, Class Hierarchies, Aggregation, Conceptual Design with ER model, Case study: ER design for Large Enterprises

**Unit II - Relational Algebra and Calculus**

Introduction to the relational model, Logical Database Design- ER to Relational, Relational Algebra - Selection and Projection, Set operations, Renaming, joins, Examples of Relational Algebra Relational Calculus- Tuple relational Calculus, Domain relational calculus.

**Introduction to Structured Query Language**

Form of Basic SQL Query, Examples of Basic SQL Queries, Introduction to Nested Queries, Correlated Nested Queries, Set Comparison Operator-Aggregate Operators, NULL values and Comparison using Null values, Logical connectivity’s – AND, OR and NOT, OUTER Joins, Disallowing NULL Values.

**Unit III - PL/SQL**

Data Types, Declaration of Variables, Strings, Control Conditional Statements, Functions, Procedures, Cursors, and Triggers

**Schema Refinement**

Introduction to schema refinement, Problems caused by decomposition, Functional dependencies (FDs) and reasoning about FDs, Normal Forms (NF) – 1NF, 2NF, 3NF and BCNF, Properties of Decomposition, Schema Refinement in Data Base Design, Case studies using Normal Forms.

**Unit IV - Transaction Management**

Transaction concept & state, Implementation of atomicity and durability, Concurrent executions of a transaction, Serializability and Recoverability, Implementation of Isolation, Testing for serializability, Lock-Based Protocols, Graph-Based Protocol, Timestamp-Based Protocols, Validation-Based, Protocols, Multiple Granularity.

**Unit V – Database File Organization and Recovery**

**Data Base File Organization**

Data on External storage, File Organization and Indexing, Cluster Indexes, Primary, and secondary indexes, Index data structures, Hash-based indexing - Static hashing and Extensible Hashing, Tree based indexing - Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index structure.

**Database Recovery**

Recovery and Atomicity, Log-based Recovery, and Recovery with the concurrent transaction.

**Course Outcomes**: At the end of the course, the students should be able to

|  |  |
| --- | --- |
| CO 1 : | Demonstrate an understanding of database management system components and features. Design E-R Model to represent real-world database application scenarios |
| CO 2 : | Demonstrate a mathematical approach towards querying a database using relational algebra and relational calculus and implement using SQL |
| CO 3 : | Convert E-R Model to a relational Model and design a proper relational database while eliminating anomalies |
| CO 4 : | Demonstrate the role of transaction management and concurrency control protocols. |
| CO 5 : | Demonstrate an understanding of database file organization and recovery of the database in case of crashes. |

**Textbooks:**

1. Database System Concepts, A.Silberschatz, H.F. Korth, S.Sudarshan, 6th edition, McGraw-Hill, 2006.
2. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, 3rd edition, TMH, 2003.

**References:**

1. Fundamentals of Database Systems, Ramez Elmasri, Shamkant B.Navathe, 7th edition, Pearson Education, 2008.
2. Database Systems: The Complete Book, Hector Garcia- Molina, Jeffery D.Ullman, Jennifer Widom, 2nd Edition, Pearson Education, 2008.
3. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, 2nd edition, PHI, 2013.

Course Code: 22HS207

**INTRODUCTION TO INNOVATION, IP MANAGEMENT AND ENTREPRENEURSHIP**

**(For B.Tech CSBS-II Year I Semester)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods /week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To understand the concepts of creativity and innovation |
| 2 : | To get awareness about intellectual property rights |
| 3 : | To identify the qualities of entrepreneur and the process of entrepreneurship |
| 4 : | To understand the components of business plan |
| 5 : | To provide insights to start a new venture |

**Unit I – Creativity and Innovation**

Concept of creativity, characteristics of creativity, types of creativity factors affecting creativity, process of creativity, techniques of creativity, sources of innovation, types of innovation, design thinking and its importance, Design thinking process

**Unit II - Introduction to Intellectual property**

Introduction, types of intellectual property, importance of intellectual property rights, Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer, intellectual property audits.

**Unit III - Entrepreneurial Perspectives**

Introduction to Entrepreneurship, Entrepreneur and Functions of entrepreneur - Traits of an Entrepreneur, Entrepreneurial Decision making Process, Role of Entrepreneurship in Economic Development, Social responsibility of Entrepreneurs, Domestic and Overseas Business OpportUnities.

**Unit IV - Business Plan**

Sources of new Ideas, Methods of generating ideas, Business Plan and its significance, steps in writing a Business Plan, Business model canvas, New Product Development Process

**Unit V - New Venture Creation**

Forms of Business Registration, Joint Ventures, Features and Objectives/Motives behind the Joint Ventures, Merger: Importance and Problems of Merger, Acquisition or Takeover, Difference between Merger and Acquisition, Franchising, Advantages and Disadvantages of Franchising, Initial public offering (IPO), IPO Process.

**Course Outcomes:** At the end of the course, the student should be able to

CO 1: Design and manage creative processes that elicit innovative solutions.

CO 2: Understand the importance of intellectual property rights and gain knowledge about IP management.

CO 3: Generate, qualify and evaluate ideas based on relevant criteria in order to create (economic, cultural, social) value in new and changing contexts

CO 4: Evaluate operational strategies for running a business with the given alternatives

CO 5: Use her or his insights into managing resources for innovation and entrepreneurship process

**Textbooks:**

1. Entrepreneurship,Robert Hisrich and Michael Peters, 10th Edition, McGraw-Hill, 2016
2. Design Thinking: Understand– Improve–Apply, Hasso Plattner, Christoph Meinel and Larry Leifer (eds), Springer, 2011.
3. Innovation and Entrepreneurship,Drucker, P.F, 1st Edition, Routledge, 2014.
4. Intellectual property right, Deborah E. Bouchoux, Cengage learning, 2012.

**References:**

1. Entrepreneurship, Arya Kumar, 4th Edition, Pearson, 2015.
2. The Dynamics of Entrepreneurial Development and Management, Vasant Desai, Himalaya Publishing House, 2015.
3. Entrepreneurial Management, Robert J. Calvin, TMH, 2004.
4. The Entrepreneurial Connection, Gurmeet Naroola, TMH, 2001.
5. The Design of Business: Why Design Thinking is the Next Competitive Advantage, Roger Martin, Harvard Business Press, 2009.
6. Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School,Idris Mootee, JohnWiley & Sons, 2013.
7. Creativity and Innovation in Entrepreneurship, Khanka SS, 1st Edition, Sultan Chand & Sons, 2021.

Course Code: 22HS208

**OPERATIONS RESEARCH**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods /week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To develop operational research models from the verbal description of the real system |
| 2 : | To model formulation and applications that are used in solving business decision problems |
| 3 : | To understand the mathematical tools that are needed to solve optimization problems |
| 4 : | To develop queuing models for optimization problems in decision making |
| 5 : | To simulate and understand the networking techniques for optimizing the cost and time in real systems |

**Unit I – Introduction to Operations Research**

Origin of OR and its definition. Concept of optimizing performance measure, Types of OR problems, Deterministic vs. Stochastic optimization, Phases of OR problem approach – problem formulation, building mathematical model, deriving solutions, validating model, controlling and implementing solution.

**Unit II – Linear Programming and Simplex Algorithm Linear Programming**

Linear programming–Examples from industrial cases, formulation. Geometric Method: 2-variable case, Special cases – infeasibility, unboundedness, redundancy &degeneracy.

Simplex Algorithm**:** slack, surplus & artificial variables, computational details, big-M method, identification and resolution of special cases through simplex iterations.

Duality – formulation, results, fundamental theorem of duality, dual-simplex and primal-dual algorithms.

**Unit III - Transportation and Assignment Problems Transportation Problems**

Examples, Definitions – decision variables, supply & demand constraints, formulation, Balanced & unbalanced situations, Solution methods – NWCR, minimum cost and VAM, test for optimality (MODI method), degeneracy and its resolution.

Assignment Problems: Examples, Definitions – decision variables, constraints, formulation, Balanced &unbalanced situations, Solution method – Hungarian, test for optimality.

**Unit IV – PERT and Inventory Control**

PERT – CPM: Project definition, Project scheduling techniques – Gantt chart, PERT & CPM, Determination of critical paths, Estimation of Project time and its variance in PERT using statistical principles, Concept of project crashing/time-cost trade-off.

Inventory Control: Functions of inventory and its disadvantages, ABC analysis, Concept of inventory costs, Basics of inventory policy (order, lead time, types), Fixed order-quantity models – EOQ, POQ & Quantity discount models.

**Unit V – Queuing Theory and Simulation Methodology Queuing Theory**

Definitions – queue (waiting line), waiting costs, characteristics (arrival, queue, service discipline) of queuing system, queue types (channel vs. phase). Kendall’s notation, Little’s law, steady state behavior, Poisson’s Process & queue, Models with examples - M/M/1 and its performance measures; M/M/m and its performance measures; brief description about some special models.

**Simulation Methodology**: Definition and steps of simulation, random number, random number generator, Discrete Event System Simulation – clock, event list, Application in Scheduling, Queuing systems and Inventory systems.

**Course Outcomes**: At the of the course, the student should be able to

|  |
| --- |
| CO 1 : Evaluate the Problems using Linear Programming |
| CO 2 : Build and solve Transportation Models and Assignment Models |
| CO 3 : Design new simple models, like CPM, PERT to improve decision – making and develop critical thinking and objective analysis of decision problems |
| CO 4 : Build and solve the problems using Inventory, Queuing theory problems |
| CO 5 : Build and solve the problems using simulation problems |

**Textbooks:**

1. Operations Research, J.K Sharma, 5th Edition, Macmillan India Pvt.Ltd, 2008.
2. Operations Research, R. Pannerselvam, 3rd Edition, Prentice Hall International, 2006.
3. Operations Research, S. D. Sharma, 2020th Edition, Kedarnath Ramnath, 2014.
4. Introductory Operations Research: Theory and Applications,Harvir Singh, Krishnadev Kumar, 7th Edition, Springer Berlin Heidelberg, 2013.

**References:**

1. Operations Research, A. M. Natarajan, P. Balasubramani, A. Tamilarasi, 4th Edition, Pearson Education, 2009.
2. Operations Research: Methods and Problems, Maurice Saseini, Arthur Yaspan and Lawrence Friedman, 5th Edition, Wiley, 1959.
3. Operations Research: An introduction, Hamdy A. Taha, 10th Edition, Pearson Education, 2019.

Course Code: 22CB231

**JAVA BASED OBJECT ORIENTED PROGRAMMING LABORATORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/Week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 1.5 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To understand OOP principles |
| 2 : | To understand the Exception Handling mechanism |
| 3 : | To understand Java collection framework |
| 4 : | To understand multithreaded programming |
| 5 : | To understand java database connectivity in Java |

**LIST OF PROGRAMS:**

**Task 1:**

1. Installation of Java software, study of any integrated development environment, Use Eclipse or Netbean platform and acquaint with the various menus. Create a test project, add a test class and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with java program to find prime numbers between 1 to n.
2. Write a Java program that prints all real solutions to the quadratic equation ax2+bx+c. Read in a, b, c and use the quadratic formula.
3. Write a Java program to multiply two given matrices.

**Task 2:**

1. Write Java program on use of inheritance, preventing inheritance using final, abstract classes.
2. Write Java program on dynamic binding, differentiating method overloading and overriding.
3. Develop a java application to implement currency converter (Dollar to INR. EURO to INR, Yen) using Interfaces.

**Task 3:**

1. Write a Java program to create a package named "com.mycompany.math" that contains a class named "Calculator" with methods to add, subtract, multiply and divide two numbers. Write a test program to use this package.
2. Create a package named "com.mycompany.util" that contains a class named "StringUtils" with a method named "reverseString" that takes a string as input and returns the reverse of the input string. Write a test program to use this package.

**Task 4:**

1. Write a Java program to implement user defined exception handling.
2. Write a Java program to throw an exception “Insufficient Funds” while withdrawing the amount in the user account.
3. Write a Java program to implement Try-with Resources, Multi-catch Exceptions, and Exception Propagation Concepts?

**Task 5:**

1. Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part where n is the sequence number of the part file.
2. Write a Java program that reads a file name from the user, displays information about whether the tile exists, whether the file is readable, or writable. The type of tile and the length of the file in bytes.

**Task 6:**

1. Write a Java program on Random Access File class to perform different read and write operations.
2. Create a class called Employee with properties name(String), dateofbirth (java.util.Date), department(String), designation(String) and Salary(double). Create respective getter and setter methods and constructors (no-argument constructor and parameterized constructor) for the same. Create an object of the Employee class and save this object in a file called “data” using serialization. Later using deserialization read this object and prints the properties of this object.

**Task 7:**

1. Create a generic class called Box that can hold any type of object. Implement the following methods: 1) void set(T obj): sets the object stored in the box 2) T get(): retrieves the object stored in the box 3) boolean isEmpty(): returns true if the box is empty, false otherwise.
2. Implement a generic Stack class that can hold any type of object. Implement the following methods: 1) void push(T obj): pushes an object onto the top of the stack, 2) T pop(): removes and returns the object at the top of the stack 3) boolean isEmpty(): returns true if the stack is empty, false otherwise.

**Task 8:**

1. Write a Java program to implement Autoboxing and Unboxing?
2. Write a Java program to implement Built-In Java Annotations?

**Task 9:**

1. Write a Java program that creates three threads. First thread displays —Good Morning every one second, the second thread displays —Hello every two seconds and the third thread displays —Welcome every three seconds.
2. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.

**Task 10:**

1. Write a Java program to create a Vector and add some elements to it. Then get the element at a specific index and print it.
2. Write a Java program to create a BitSet and set some bits in it. Then perform some bitwise operations on the BitSet and print the result.
3. Write a Java program to read the time intervals (HH:MM) and to compare system time if the system Time between your time intervals print correct time and exit else try again to repute the same thing. By using String Tokenizer class.

**Task 11:**

1. Write a Java program to implement Generic stack using Array List Collection class.
2. Write a Java program to implement Generic stack using LinkedList Collection class

**Task 12:**

1. Write a Java program to implement Generic queue using Array List Collection class.
2. Write a Java program to implement Generic queue using LinkedList Collection class.

**Task 13:**

1. Write a Java program to demonstrate the working of different collection classes. [Use package structure to store multiple classes].
2. Write a Java program to create a TreeMap and add some elements to it. Then get the value associated with a specific key and print it.
3. Write a Java program to create a PriorityQueue and add some elements to it. Then remove the highest priority element from the PriorityQueue and print the remaining elements.

**Task 14:**

1. Deve1op a Java application to establish a JDBC connection, create a table student with attributes-name, register number, mark 1, mark2, mark3. Insert the values into the table by using the java and display the information of the students at font end.
2. Write a program to perform CRUD operations on the student table in a database using JDBC

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Able to write the programs for solving real world problems using Java OOP principles |
| CO 2 : | Able to write programs using Exceptional Handling approach |
| CO 3 : | Able to write multithreaded applications |
| CO 4 : | Able to Build application using Java Collection Framework |
| CO 5 : | Able to develop java application connect database using JDBC |

**References:**

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education, 2014.
2. Thinking in Java, Bruce Eckel, 4th Edition Pearson Education, 2006
3. Understanding Object-Oriented Programming with Java, T. Budd, 1st edition, Pearson Education, 2000
4. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning, 2003
5. Core Java, Cay S. Horstmann and G Cornell, Volume 1, 9th edition, Pearson Education, 2013
6. Programming in Java, S. Malhotra, S. Chaudhary, 2nd edition, Oxford Univ. Press, 2018
7. Java Programming and Object-oriented Application Development, R. A. Johnson, First Edition, Cengage Learning, 2016

**Online resources:**

1. https://www.w3schools.com/java/java\_oop.asp
2. http://peterindia.net/JavaFiles.html

Course Code: 22CS232/22CS282

**DATABASE MANAGEMENT SYSTEMS LAB**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 2 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 1.5 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To understand the relational model |
| 2 : | Analyze database requirements and determine the entities involved in the system and their relationship to each other |
| 3 : | Understand logical design of the database modeling concepts such as E-R diagrams |
| 4 : | Demonstrated SQL DML/DDL commands to insert and manipulate the database. |
| 5 : | Understand procedures, functions and triggers in PL/SQL |

**Database Description:** This lab enables the students to practice the concepts learnt in the subject DBMS by developing a database for an example ― **Boat reservation by the sailor** ‖and ―**employee data maintenance in an organization**‖ whose description is as given below. The student is expected to practice the designing, developing and querying a database in the context of reserving a boat and employee data maintenance. Students are expected to use ― MySql database.

**"Boat reservation by the sailor"** is a schema with several boats which could be reserved depending on color and availability on a particular day. The sailor reserves the boat on a particular day y registering himself with a rating. The sailor is identified by sailor id, boats are identified by boat id and reservation is uniquely identified by sailor id, boat id and day.

**“Employee data maintenance in an organization”:** In any organization, we need to maintain the data of employees categorized into department as per the salary. The scheme contains employee, department and sal grade tables which are identified by employee id, department id and range of salary respectively.

1. **E-R Model**

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc. Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Boat reservation by the sailor:

Entities:

1. SAILORS

2. BOATS

3. RESERVES

PRIMARY KEY ATTRIBUTES:

1. SID (SAILOR ENTITY)

2. BID (BOATS Entity)

3. SID,BID,DAY (RESERVES ENTITY)

Employee data maintenance in an organization Entities:

1. EMPLOYEE

2. DEPT

3. SALGRADE

PRIMARY KEY ATTRIBUTES:

1. EID (EMPLOYEE ENTITY)

2. DID (DEPT Entity)

3. LOWSAL AND HIGHSAL (SALGRADE ENTITY)

1. **Concept design with E-R Model**

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc. wherever required for

1) Boat reservation by the sailor

2) Employee data maintenance in an organization

1. **Relational Model**

1) SELECT - retrieve data from the a database

2) INSERT - insert data into table

3) UPDATE - updates existing data within a table

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi valued, and Derived) have different way of representation.

SAILORS

SID SNAME RATING AGE

EMPLOYEE

EID ENAME DID SAL DESIGNATION MGRNUM DOJ AGE

**4. Normalization**

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

Perform do the second and third normal forms for sailors and Employee databases if required.

**5. Installation of Mysql and practicing DDL commands**

Installation of MySql. In this week student will learn Creating databases, How to create tables, altering the database, dropping tables and databases If not required. Students will also try truncate, rename commands etc.

**6. Practicing DML commands**

DML commands are used to for managing data within schema objects. Some examples:

4) DELETE - deletes all records from a table, the space for the records remain

**7. Querying - I**

In this week students are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

1. **Querying - II**

Students are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

**9. Triggers**

In this week students are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

**10. Procedures**

In this session students will learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

**11. Cursors**

In this week students will learn to declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done.

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Analyze database requirements and determine the entities involved in the system and their relationship to each other |
| CO 2 : | Design E-R Model to represent database application scenarios |
| CO 3 : | Convert/transform the E-R Model to relational tables, populate relational database and formulate SQL queries on data |
| CO 4 : | Improve the database design by normalization. |
| CO 5 : | Implement PL/SQL procedures, function, triggers and cursors |

**References:**

1. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, 3rd edition, TMH, 2003.
2. Introduction to SQL, 4th edition, Rick F.VanderLans, Pearson education, 2007.
3. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, 2nd edition, Pearson education, 2002

Course Code: 22HS232

**OPERATIONS RESEARCH LABORATOTRY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 2 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 1 | Semester End Exam Duration | : | 2 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To understand the methodology of operations research and execute the output with TORA Software |
| 2 : | To understand the concept of graphical analysis of LPP |
| 3 : | To identify a transportation problem and implement various transportation model |
| 4 : | To derive the solution of an assignment problem and integer programming problems using various methods |
| 5 : | To understand Classification of Queuing Models ( Finite and Infinite Models) and the simulation models |

**WEEK 1, 2, 3:**

**Experiment 1:** To solve Linear Programming Problem using Graphical Method with

I. Unbounded solution

II. Infeasible solution

III. Alternative or multiple solutions.

**WEEK 4, 5, 6:**

**Experiment 2:** Solution of LPP with simplex methods.

**WEEK 7, 8, 9:**

**Experiment 3:** Solving transportation problems.

**WEEK 10, 11, 12:**

**Experiment 4:** solving assignment problems.

**WEEK 13,14,15,16:**

**Experiment 5:** Solving queuing and simulation models

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Laboratory problems executed b by TORA software. |
| CO 2 : | Give solution of any LPP by using simplex algorithm. |
| CO 3 : | Implement feasible, basic, non-degenerate solutions of a transportation problem |
| CO 4 : | Implement assignment problem that will be balanced and also solve integer Programming problem. |
| CO 5 : | Implement queuing theory techniques and simulation models. |

**Textbooks:**

1. Operations Research: An introduction, Hamdy A. Taha, 10th Edition, Pearson Education, 2019.
2. Principles of OR with Application to Managerial Decisions, H. M. Wagner, 2nd Edition, Prentice Hall India Learning Private Limited, 1980.
3. Introduction to Operations Research, F. S. Hiller and G. J. Lieberman, 10th Edition, McGraw Hill Education, 2017.
4. Operations Research, S. D. Sharma, 2020th Edition, Kedarnath Ramnath, 2014.

**References:**

1. Linear Programming, K. G. Murthy, 1st Edition, John Wiley & Sons Inc, 1983.
2. Elements of Queuing Theory, Thomas L. Saaty, 11th Edition, Dover Publications Inc., 1983.
3. Operations Research and Management Science, A. Ravi Ravindran, Hand Book, 1st Edition, CRC Press, 2007.
4. Management Guide to PERT/CPM, Wiest & Levy, 2nd Edition, Prentice Hall India Learning Private Limited, 1979.
5. Modern Inventory Management, J.W. Prichard and R.H. Eagle, 2nd Edition, Wiley & Sons, 1965.

Course Code: 22HS233

**BUSINESS COMMUNICATION AND VALUE SCIENCE –III**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 2 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 1 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To introduce Self-analysis techniques like SWOT & TOWS. |
| 2 : | To introduce key concepts of pluralism, cultural spaces, and cross-cultural communication. |
| 3 : | To understand the importance of ancient knowledge of science for nation development. |
| 4 : | To introduce mechanics, features, and style of technical writing. |
| 5 : | To create clear, accurate, and succinct content to SOP, and Technical Reports for academics as well as for workplace. |

**Task 1**

1. **SWOT (Strength, Weakness, OpportUnity & Threat)**

**Task 2**

**Suggested activity- Just a minute**

1. Pluralism - Cultural & Gender awareness
2. Cross-cultural communication- Culture shock.

**Task 3**

**Suggested activity- Group Discussion**

1. Ancient & Modern Science & Technology, Knowledge System

**Task 4**

**Suggested activity-Presentation**

1. Artificial intelligence and Ethics, Human communication and interference of AI.

**Task 5**

**Suggested activity- Storytelling**

1. Technical Writing - rules features, mechanics of technical writing, errors in technical writing.

**Task 6**

**Suggested activity- Worksheets**

1. Statement of Purpose- purpose, types, format & process

**Task 7**

**Suggested activity- SoP Submission**

1. Summarizing & Synthesizing- method & techniques

**Task 8**

**Suggested activity- Listening to Speeches and Public Speaking**

1. Abstract Writing

**Task 9**

**Suggested activity- Abstract Submission**

1. Report Writing – informative & analytical reports format

**Task 10**

**Suggested activity- Report Submission**

1. Business Proposal.

**Task 11**

**Suggested activity- Proposal Submission**

**Task 12**

**Project Work:**

Visit rural area/ underprivileged parts of city to address some of the local issues; if relevant, suggest a practical technological solution to the issues.

**Course Outcomes:** At the end of the course, the student will be able to

|  |  |  |
| --- | --- | --- |
| CO 1 | : | Apply and analyze the basic principles of SWOT and life positions |
| CO 2 | : | Identify and respect pluralism, analyze cross cultural communication globally, and recognize roles and relations of different genders. |
| CO 3 | : | Apply ancient knowledge of science for nation development |
| CO 4 | : | Learn the best practices of technical writing and understand, apply, and analyze the tools of technical writing. |
| CO 5 | : | Apply the basic communication practices of formal writing. |

**References:**

1. Effective Technical Communication, Ashraf Rizvi M., 2nd Edition, Tata McGraw Hill Publishing Company Limited, New Delhi, 2005.
2. Technical Communication: A Practical Approach, William S. Pfeiffer, 8th Edition, Pearson, 2012.
3. Technical Communication, Burnett, Rebecca, 6th Edition, Cengage Learning, 2001.
4. SWOT Analysis: A Guide to SWOT for Business Studies Students, Alan Sarsby, 2016.
5. Cultures and/or Globalization,Barrie Axford, Richard Huggins, Cambridge Scholars Publishing, 2011.
6. Responsible AI and Ethical Issues for Businesses and Governments, Bistra Vassileva, Moti Zwilling, 2020.

Course Code: 22HS211

**HAPPINESS AND WELLBEING**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instruction | : | 2 Periods/week | Continuous Internal Evaluation | : 100 marks |
| Tutorial | : | - | Semester End Examination | : - |
| Credits | : | - | Semester End Exam Duration | : - |

**Course Objectives:**

|  |  |  |
| --- | --- | --- |
| 1 | : | To explore diverse definitions of happiness and understand its function in everyday life. |
| 2 | : | To learn the various perspectives of happiness. |
| 3 | : | To understand the meaning of life and live for well-being of self & society. |
| 4 | : | To strike a balance between natural and psychological stress. |
| 5 | : | To devise strategies to cope with physical and mental disorders and nurture positive attitude. |

**Unit I - Concept & factors of Happiness**

Operationalizing happiness: Definition of happiness, Understanding the construct of happiness.

Factors influencing happiness: Biological, psychological & socio-cultural factors, case studies and self-learning Tasks. Environmental and Social, Physical, emotional, and psychological well-being for happiness, physiological and hormonal basis of happiness

**Unit II - Various myths & perspectives of happiness**

The myths as well as the simple secrets of Happiness: the key ingredients with scientific evidence, case studies and online assignments and assessments. Sustainable model of happiness

Models of happiness :  Humanistic perspective of Carl Roger and Maslow. lndological Theories of Happiness: Panch Kosh Theory & Idea of Well-Being, Idea of Self and other, Hierarchy and stages of happiness

**Unit III - Art of mindfulness**

Train yourself in Happiness: [Happiness & body language](https://archive.nptel.ac.in/courses/109/105/109105199/), [Innovation & Happiness](https://archive.nptel.ac.in/courses/109/105/109105199/), [Conflicts and Stress](https://archive.nptel.ac.in/courses/109/105/109105199/), [Stress and Coping](https://archive.nptel.ac.in/courses/109/105/109105199/), [Reconnecting happiness: beyond pandemic](https://archive.nptel.ac.in/courses/109/105/109105199/), [Meditation and Mindfulness: An Overview](https://archive.nptel.ac.in/courses/109/105/109105199/), [Creativity, Intuition and Flow](https://archive.nptel.ac.in/courses/109/105/109105199/)

**Unit IV -** **Happiness: Cross-cultural Contexts**

Culture and Happiness, Interpersonal Relationship: Comparative Perspective, Towards Self-Actualization

Local and Global Perspective of Happiness: Measuring happiness: Key indicators, Happiness Index, India in Global Happiness Indices

**Unit V - Coping strategies**

Unconscious mind and defense coping strategies

Characteristics of Constructive coping, physical ways of coping & mental ways, coping & social support, coping with meditation & mindfulness,

Resilience Meaning, Nature and Approaches, Theories of Resilience, Promoting Resilience, Resilience in the phase of challenge & loss of Empathy and Altruism

Positive Emotions and well-being: Hope & Optimism, Love the Positive, Psychology of Emotional Intelligence, Influence of Positive Emotions

Strengths and Virtues of Character

Forgiveness and Gratitude, Personal transformation and Role of suffering, Trust and Compassion

**Course Outcomes:** At the end of the course, students will be able to

|  |  |  |
| --- | --- | --- |
| CO1 | : | Identify the socio-demographic factors of happiness. |
| CO2 | : | Develop tools to strengthen their approach towards self and society. |
| CO3 | : | Programme their mind to focus on necessary issues and channelize their energy. |
| CO4 | : | Relate to the global phenomenon of sustainable development and become sensitive to the needs of the planet. |
| CO5 | : | Recognize the importance of striking balance between mind, body, and commUnity to manage emotions and behaviors for greater happiness. |

**Recommended activities:**

1. Community surveys on the facilities promoting positive mental health.
2. Practices such as Yoga and Meditation
3. Visit retreat centers, recreation clubs, and parks.
4. Extending help and social service by visiting old age homes/ hospitals/slum areas or any other disadvantaged groups. Students can undertake a field work/ project independently or work as an intern with NGOs working in happiness and well-being.
5. Critical appreciation of a documentary/ film based on Happiness and Wellbeing can be undertaken by the students.
6. Workshops/ Sessions for the actualization of innate creative potential- (Music, Drawing, Calligraphy, Dramatics)
7. Hands-on Happiness: Gardening, Cleaning, Washing, Cooking, etc.
8. If required, students can share their experiences in the form of a Project Report.
9. Students may share their experiences in the form of Audio-video presentations of 15-20 minutes.

**References:**

1. Positive Psychology, S Baumgardner & M. Crothers, New Delhi: Pearson Education, India, 2014.
2. Social Intelligence: The New Science of Human Relationships, D. Goleman, Bantam Dell, A Division of Random House, Inc., New York, 2007.
3. Pursuits of Happiness, Mathews, Gordon and Carolina Izquierdo (eds), Berghahn Books, New York, 2010.
4. The Synthesis of Yoga, Sri Aurobindo, Part Three: The Yoga of Divine Love, Chapter 7, The Ananda Brahman, Sri Aurobindo Ashram, Pondicherry, pp. 569-570.
5. Happiness Unlimited: Awakening with BRAHMA KUMARIS, B. K. Shivani, 1st Edition, Amaryllis, 15 June 2019.

Course Code: 22CB251

**PYTHON FOR EXPLORATORY DATA ANALYSIS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To impart python programming skills for selecting effective data structures |
| 2 : | To enable data manipulation and file handling proficiency |
| 3 : | To offer skills in protecting the program from unwanted errors |
| 4 : | To gain mastery of Linear Algebra functions over high dimensional data |
| 5 : | To present data cleaning and manipulation functions |

**Unit I – Introduction**

Basic Data Types, variables, statements, expressions, Operators, Strings, Control Structures- Branching and looping structures.

**Data Structures**

Lists, Tuples, Sets, and Dictionaries

**Unit II - Functions and File Handling**

Data Link Layer Design Issues, Function Definition, Function Call, Variable Scope and Lifetime, return statement, More on Defining Functions, Lambda Functions or Anonymous Functions, Documentation Strings, Recursive Functions, Modules and Packages in Python, Standard Library modules, Globals(), Locals(), and Reload(). Functional programming – map(),filter(),reduce()

Opening and Closing Files, Reading and Writing Files, File positions,

**Unit III -** **Advanced Features of Python and Object-Oriented Programming**

Exception Handling, Comprehension Syntax, Regular Expression – Syntax, special characters

Classes and Objects, Creating Classes in Python, Creating Objects in Python, The Constructor Method, Classes with Multiple Objects, Class Attributes versus Data Attributes, Encapsulation, Inheritance, Polymorphism

**Unit IV – Data Science Tools Kits: Numpy and Pandas**

**NumPy Basics:** Arrays and Vectorized Computation. The NumPy ndarray, Universal Functions: Fast Element-Wise Array Functions, Array-Oriented Programming with Arrays, File Input and Output with Arrays, Pseudorandom Number Generation, Advanced Array Manipulation, Reshaping Arrays, Concatenating and Splitting Arrays, Repeating Elements: tile and repeat, Fancy Indexing Equivalents: take and put, Broadcasting.

**Pandas Basics:** Introduction to pandas Data Structures: Series. Data Frame, Index Objects, Essential Functionality, Summarizing and Computing Descriptive Statistics,

**Unit V – Data Analysis and Visualization in Python**

Data Wrangling: Join, Combine, and Reshape, Data Aggregation and Group Operations, Time Series, Categorical Data, Advanced GroupBy Use.

**Matplotlib:** Plotting and Visualization: Figures and Subplots, Line Plots, Bar Plots, Histograms and Density Plots, Scatter or Point Plots.

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Gain mastery over the syntax and data structures of Python that matches the industry needs |
| CO 2 : | Design and implement reusable code modules and adopt functional programming skills |
| CO 3 : | Demonstrate the polymorphism and overloading features of Python |
| CO 4 : | Prepare the data ready for next steps in data pipeline by removing noisy and imputing missing elements and derive better insights on statistical properties |
| CO 5 : | Conduct visual analytics and gain deeper understanding on data characteristics |

**Textbooks:**

1. Python Programming Using Problem Solving Approach, Reema Thareja, Oxford University Press, 2017.
2. Python for Data Analysis, McKinney, Second Edition, 2017.

**References:**

1. Data Science from Scratch: First Principle with Python, Joel Grus, First Edition, O’Reilly, 2017.
2. Programming in Python 3: A Complete Introduction to Python Language, Mark Summerfield, Second Education, Addison-Wesely, 2009.

Course Code:22IT252

**OPERATING SYSTEMS**

(Common to CSE, CSBS, CSE -AI&ML, CSE-CS, CSE-DS and IT)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration | : | 3 Hours |

**Prerequisites:**

1. A course on “Computer Programming and Data Structures

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection) |
| 2 : | Introduce the issues to be considered in the design and development of operating system |
| 3 : | Introduce basic Unix commands, system call interface for process management, inter-process communication and I/O in Unix |

**Unit I – Operating System Introduction**

Operating System - Introduction, Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads. Process related system calls – fork, exit, weight and exec.

**Unit II- CPU Scheduling, Process Management and Synchronization**

CPU Scheduling - Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling.

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware and Software, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors

**Unit III- Interprocess Communication Mechanisms and Deadlocks**

Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

**Unit IV- Memory Management and Virtual Memory**

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

**Unit V- File System Interface and Operations**

File System Interface and Operations -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Disk scheduling algorithms, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

**Course Outcomes**: At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Understand the role of Operating System with its function and services. |
| CO 2 : | Compare various algorithms used for CPU scheduling and apply various concepts related to concurrency and synchronization to solve problems |
| CO 3 : | Understand the inter process communication mechanism and resolve deadlock in a multi-programmed environment. |
| CO 4 : | Understand the concepts of virtual memory and how it is realized in systems. |
| CO 5 : | Differentiate and Demonstrate file systems, directory structures and their implementation issues. |

**Textbooks:**

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 8th edition, John Wiley, 2006.
2. Advanced programming in the UNIX environment, W.R. Stevens, Pearson education, 2013.

**References:**

1. Operating Systems- Internals and Design Principles, William Stallings, 5th Edition, Pearson Education/PHI, 2005
2. Operating System A Design Approach, Croley, TMH, 2017.
3. Modern Operating Systems, Andrew S. Tanenbaum, 2nd Edition, Pearson/PHI,2016
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education, 2015

Course Code: 22DT251

**Design and Analysis of Algorithms**

(Common to CSBS, CSE-AI&ML, CSE-CS, CSE-DS)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |  |
| --- | --- | --- |
| 1 | : | To emphasize upon the demands of real-world problems in engineering solutions |
| 2 | : | To make students conversant with the various paradigms of algorithms |
| 3 | : | To handcraft the performance analysis of designed solutions |
| 4 | : | To take students through various optimization principles of ill-posed problems |

**Unit I – Fundamentals of algorithm analysis**

Introduction- Definition of algorithm, algorithmic problem solving, pseudo code for expressing algorithms. Asymptotic notations- ο, Ω, and θ notations. Performance analysis: Time and space complexity: count, tabular methods, examples on non-recursive, recursive algorithms. Recursive algorithms and recurrence relations - ToH problem, Amortized analysis.

**Unit II** - **Algorithm paradigms: Divide and Conquer**

Control abstraction, binary search algorithm and its complexity, Merge sort, its complexity, quick sort, its complexity. Graph traversals: Depth first search (DFS), breadth first search (BFS), articulation points, bi-connected components.

**Unit III** - **Greedy paradigm**

Control abstraction, fractional knapsack problem, job sequencing problem, minimum cost spanning tree: Prim’s algorithm, Kruskal’s algorithm, Dijkstra’s algorithm.

**Unit IV - Dynamic programming and Backtracking**

**Dynamic programming -** Control abstraction, Multistage Graphs, OBST, Travelling salesperson problem, reliability design, 0/1 knapsack problem, All pair shortest path algorithm.

**Backtracking -** n-queens problem, Graph coloring, Sum of subsets problem.

**Unit V** - **Branch and Bound, Complexity Theory**

**Branch and Bound:** General method, Applications: Travelling salesperson problem, 0/1 knapsack problem, LC branch and bound solution, FIFO branch and bound solution, Game trees.

**Np-hard and NP-complete problems:** basic concepts, non-deterministic algorithms, NP-hard and NP-complete classes, Cook’s theorem.

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |  |
| --- | --- | --- |
| CO 1 | : | Analyze worst-case running times using asymptotic analysis of algorithms. |
| CO 2 | : | Describe the divide-and-conquer paradigm and Synthesize divide-and-conquer algorithms. |
| CO 3 | : | Define optimization problems and solve them through various greedy policies |
| CO 4 | : | Describe the dynamic-programming paradigm and synthesize dynamic-programming algorithms and analyze them. |
| CO 5 | : | Reduce the size of search space of the optimization problems by applying backtracking and branch and bound tools. Appreciate the Non-Deterministic modeling of algorithms. |

**Textbooks:**

1. Fundamentals of algorithms, E. Horowitz and S.Sahni, 2nd edition, Galgotia Publications, 2010
2. Introduction to algorithms, T.H.Cormen, C.E.Leiserson, R.L.Rivest, and C.Stein, 2nd edition, PHI/Pearson Education, 2001.

**References:**

1. Introduction to Design and Analysis of Algorithms, R C T Lee, Hang and TT Sai, A strategic approach, TMH.

2. Data structures and Algorithm Analysis in C++, Allen Weiss, 2nd edition, Pearson Education

3. Design and Analysis of Algorithms, Aho, Ullman and Hopcroft, Pearson Education, 1974.

4. Algorithms, Richard Johnson Baugh, and Marcus Schaefer, Pearson Education.

Course Code: 22CB252

**PRINCIPLES OF SOFTWARE ENGINEERING**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To identify the software development activities and process models |
| 2 : | To understand the importance of Project planning and management |
| 3 : | To explore various metrics and quality assurance strategies |
| 4 : | To analyze different strategies for testing and risk management |

**Unit I**

**Introduction:** Programming in the small vs. programming in the large, software project failures. Basic concepts of software life cycle models - The Waterfall model, Spiral model and Agile methodology, XP, Scrum, concepts of feasibility study, software cost estimation models and concepts of software engineering economics.

**Unit II**

**Software Requirements:** Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

**Requirements engineering process**: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

**Unit III**

**Design Engineering:** Design process and design quality, design concepts, the design model. **Creating an architectural design**: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

**Object Oriented Analysis, Design and Construction:** Class Responsibility Collaborator (CRC) model; quality of design; design measurements; concepts of design patterns; Refactoring; object-oriented construction principles; object-oriented metrics.

**Unit IV**

**Software Testing:** Introduction to faults and failures Basic testing concepts; concepts of verification and validation; black box and white box tests;

**White Box Testing:** White box test coverage – code coverage, condition coverage, branch coverage;

**Black Box Testing**: basic concepts of black box tests –equivalence classes, boundary value tests; testing use cases; testing for non-functional requirements – volume, performance and efficiency; concepts of inspection.

**Unit V**

**Risk management:** Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM.

**Software Quality and Reliability:** Internal and external qualities; process and product quality; principles to achieve software quality; introduction to different software quality models like McCall, Boehm, FURPS / FURPS+, Dromey, ISO – 9126; introduction to Capability Maturity Models (CMM and CMMI); introduction to software reliability, reliability models and estimation. Formal technical reviews, statistical software quality assurance.

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Analyze software engineering framework activities and process models that can be tailored with appropriate methods to manage risk. |
| CO 2 : | Understand the role of software quality and different maturity models. |
| CO 3 : | To Understand the Requirements Engineering Process and the metrics for different models of the system using the perception of UML. |
| CO 4 : | To understand the oops concepts to analyze the use cases. |
| CO 5 : | Understand the role of testing in the software development cycle and be capable of developing a test plan to deliver quality software. |

**Textbooks:**

1. Software Engineering, Sommerville, 10th edition, Pearson Education, 2017
2. Software Engineering, A practitioner’s Approach, Roger S. Pressman, 8th edition, McGraw-Hill International Edition, 2014.

**References:**

1. The Unified Development Process, Ivar Jacobson, Grady Booch, James Rumbaugh, 1st Edition, Pearson Education, 2002
2. Design Patterns: Elements of Object-Oriented Reusable Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides , Addison Wesley, 2009.
3. Software Metrics: A Rigorous and Practical Approach, Norman E. Fenton, Shari Lawrence, Pfleeger, CRC 3rd Edition, Press, 2014.
4. Fundamentals of Software Engineering, Carlo Ghezzi, Jazayeri Mehdi, Mandrioli Dino, Second Edition, Pearson Education, 2007.
5. Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices, Michael Jackson, Addison-Wesley Professional, 1995.

Course Code: 22HS252

**COMPUTATIONAL STATISTICS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 3 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |
| --- |
| 1. : To learn multidimensional generalization of a univariate normal random variable |
| 1. : To learn concept of multivariable linear regression model |
| 1. : To learn linear discriminant function analysis |
| 1. : To learn data Summarization and data reduction using Factor analysis |
| 1. : To learn grouping the data using Cluster analysis |

**Unit I - Multivariate Normal Distribution & Linear Regression Model**

Multivariate Normal Distribution Functions, Conditional Distribution and its relation to regression model, Standard multiple regression models with emphasis on detection of collinearity, outliers.

**Unit II - Multivariate Regression**

Assumptions of Multivariate Regression Models, Parameter estimation, Multivariate Analysis of variance and covariance.

**Unit III - Discriminant Analysis**

Statistical background, linear discriminant function analysis, Estimating linear discriminant functions and their properties.

**Unit IV - Factor Analysis**

Factor analysis model, Extracting common factors, determining number of factors, Transformation of factor analysis solutions, Factor scores.

**Unit V - Cluster Analysis**

Introduction, Types of clustering, Correlations and distances, hierarchical clustering, overlapping clustering, clustering by partitioning methods (K-Means clustering).

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Model the linear relationship between the explanatory (independent) Variables and response(dependent)variables |
| CO 2 : | Make better decisions using linear regression techniques |
| CO 3 : | Develop discriminant functions that will discriminate between the categories of the dependent variable in a perfect manner. |
| CO 4 : | Reduce a large number of variables into fewer numbers of factors using factor analysis |
| CO 5 : | Gain the Knowledge on different types of Clustering. |

**Textbooks:**

1. An Introduction to Multivariate Statistical Analysis, T.W. Anderson, 3rd Edition, Wiley, 2003.
2. Applied Multivariate Data Analysis, J. D. Jobson, 1st Edition, Springer, 1994.
3. Statistical Tests for Multivariate Analysis, Heinz Kres, 1st Edition, Springer, 1983.

**References:**

1. Regression Diagnostics, Identifying Influential Data and Sources of Collinearity, D.A.Belsey, E.Kuh and R.E.Welsch, 2nd Edition, Wiley, 1980.
2. Applied Linear Regression Models, J.Neter, W.Wasserman and M.H.Kutner, 4th Edition, McGraw-Hill Education, 2004.
3. Python for Data Analysis, Wes Mc Kinney, 2nd Edition, O′Reilly, 2017.
4. Applied Linear Regression Models, J.Neter, W.Wasserman and M.H.Kutner, 4th Edition, McGraw-Hill Education, 2004.
5. Introduction to Linear Regression Analysis, D.C.Montgomery and E.A.Peck, 3rd Edition, Wiley India Pvt Ltd, 2006.

Course Code: 22CB281

**COMPUTATIONAL STATISTICS AND EDA LABORATORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 1 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To offer deeper insights about collections feature of Python |
| 2 : | To handle large amounts of data for extracting better features |
| 3 : | To present the insights as graphs and charts |
| 4 : | To impart statistical tool and methods that give better inferences |

**LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:**

**Task 1:**

1. Write a Python program with the following functionalities:
2. Prompt the user for an integer input and store it in a variable.
3. Receive a floating-point number from the user and assign it to another variable.
4. Construct a complex number by obtaining real and imaginary parts entered by the user.
5. Interact with the user to input a Boolean value (True or False) and store it.
6. Ask the user for a string input and save it in a distinct variable.
7. Write a Python program that takes input for an integer and showcases its binary, octal, and hexadecimal representations.
8. Write a Python program to demonstrate various types of conversion functions.
9. Write a Python program to demonstrate the operators in Python.

**Task 2:**

1. Write a Python program to showcase different methods for accessing strings.
2. Write a Python program to demonstrate different string methods.
3. Write a Python program to demonstrate List methods.
4. Write a Python program to demonstrate Tuple operations.

**Task 3:**

1. Write a Python program to demonstrate Dictionary operations.
2. Write a Python program to demonstrate Set operations.

**Task 4:**

1. Write a Python function that checks the eligibility of an individual to vote based on their age. The function should take the user's age as input and return a message indicating whether the person is eligible to vote or not.
2. Write a Python function that determines whether a given number is a "Lucky Number" based on the count of its divisors. The function should take a user-inputted number, calculate its divisors, and print "Lucky Number" if the count of divisors is 4, otherwise print "Better Luck Next Time.
3. Write a Python function to calculate the shipping cost for a luggage shipment based on weight, distance, shipping speed, and insurance options.

The shipping cost is calculated as follows:

For a weight up to 10 kg, the cost is 10 Rs per kilometer

For a weight between 10 kg and 20 kg, the cost is 20 Rs per kilometer

For a weight over 20 kg, the cost is 30 Rs per kilometer

For express shipping, the cost is doubled

For standard shipping, the cost is increased by 50%

If insurance is requested (default is False), there is an additional charge of 5 Rs per kg.

**Task 5:**

1. Write a Python program that reads a text file and counts the number of occurrences of a given word.
2. Write a Python program to compare two files.
3. Write a Python program to exchange the contents of two files.

**Task 6:**

1. Write a Python program to count the number of records stored in the file employee.
2. Write a Python program to merge two files into a third file. The names of the files must be entered using command line arguments.
3. Write a Python program to read a file and count the total number of lines and words in a file.

**Task 7:**

1. Write a Python program to demonstrate the handling of standard exceptions.
2. Write a Python program to demonstrate the handling of user-defined exceptions.

**Task 8:**

1. Write a Python program to demonstrate the following:
   * Classes and Objects
   * Constructors
2. Write a Python program to unravel Polymorphism in Python – Method Overloading and Overriding.
3. Write a program to demonstrate Inheritance in Python.

**Task 9:**

1. Perform Create a NumPy array with a predefined sequence of integers. Demonstrate the process of indexing and slicing to extract specific elements or subarrays.
2. Analyze financial data using NumPy Universal functions.

**Task 10:**

1. Implement a program to perform array-based calculations efficiently using NumPy.
2. Demonstrate file input and output with Arrays
3. Implement the below essential functions of Numpy using a data set:

arrange, reshape, ravel, transpose, broadcast, squeeze, load, and save.

**Task 11:**

1. Create a Pandas DataFrame from a dictionary containing six keys and their corresponding values.
2. Utilize the dictionary to construct a Pandas DataFrame.
3. Verify the contents of the newly created DataFrame.
4. Display the first and last three rows from DataFrame.

**Task 12:**

1. Load a dataset named “students\_information.csv” into a Pandas DataFrame.
2. Inspect the top and bottom five rows of the DataFrame.
3. Access specific columns.
4. Print columns at index 2,3 and 4

**Task 13:**

1. Write a Python program to generate a Histogram to visualize the distribution of the dataset
2. Create a data frame for the following sales information of Lenovo Laptop in the year 2022 :

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

sales = [12000, 15000, 18000, 14000, 16000, 19000, 22000, 25000, 28000, 32000, 28000, 21000]

Generate a Bar plot for Monthly Sales of Lenovo Laptop Products for the Year 2022.

1. Consider the following Bug's information generated by various team members for a particular software application and Generate a Pie chart to represent it.

severity\_levels = ['High', 'Medium', 'Low'] bug\_counts = [2, 2, 2]

**Task 14:**

1. Consider a dataset containing details on students' study hours and their corresponding exam scores.

a) Visualize the relationship between years of tutorial\_hours and exam\_score using scatterplot.

b) Analyze the pattern to ascertain the type of correlation between tutorial\_hours and exam\_score.

Note: Generate synthetic data for study\_hours and exam\_scores attributes for 30 students.

**Task 15:**

1. Write a Python program to analyse data from any Kaggle dataset for performing the multivariate regression model.
2. Implement python program for analyse the variance of multivariate variables when categorical data is given for the analysis

**Task 16:**

1. Create a python program for data reduction techniques of any benchmark dataset using linear discriminant techniques
2. Write and implement python program on a business data for the analysis using factor analysis

**Task 17:**

1. Create a python program for farming the clusters in order to identify the similar objects from the marketing data set for the analysis

**Note: Tasks 1 to 14 are mandatory and Tasks 15 to 17 are optional.**

**Course Outcomes**: At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Utilize the features of Python data structures to represent real world data items |
| CO 2 : | Efficiently process the data present in flat files using arithmetic operations and regular expression based pattern matching |
| CO 3 : | Fix the data into Dataframe and perform cleaning, indexing and aggregation operations |
| CO 4 : | Offer descriptive features of data by employing statistical methods |
| CO 5 : | Visually present the trends and correlations among the features of dataset |

**References:**

1. Python Programming Using Problem Solving Approach, Reema Thareja, Oxford University Press,2017.
2. Python for Data Analysis, McKinney, Second Edition, 2017.
3. Data Science from Scratch: First Principle with Python, Joel Grus, First Edition, O’Reilly, 2017.
4. Programming in Python 3: A Complete Introduction to Python Language, Mark Summerfield, Second Education, Addison-Wesely, 2009.

Course Code: 22CB282

**ALGORITHMS AND OPERATING SYSTEMS LABORATORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 1 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | Analyze the asymptotic performance of algorithms. |
| 2 : | Demonstrate a relationship between major algorithms and data structures. |
| 3 : | To implement problems using different algorithm design paradigms and analyze their time complexity. |
| 4 : | To implement various CPU scheduling, memory management and virtual memory techniques. |
| 5 : | To implement file management and disk scheduling algorithms |

**LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:**

**Task 1:**

1. Implement and analyze run time complexity of Non-Recursive and Recursive Problem Solutions.
2. Implement and analyze run time complexity in best & worst case for Binary Search, Quick Sort
3. Implement and analyze run time complexity in best & worst case for Merge Sort.

**Task 2 :**

1. Implement and analyze run time complexity of Greedy Application Problems: Knapsack, Job Sequencing, Huffman Tree generation.
2. Implement and analyze run time complexity of Greedy Application Problems: Prims and Kruskal’s.

**Task 3 :**

1. Implement and analyze run time complexity of Dijkstra’s Single Source shortest path algorithm.
2. Implement and analyze run time complexity of DFS and its applications using Topological Sorting

**Task 4**

1. Implement and analyze run time complexity of Dynamic Programming Application Problems: Optimal binary search tree, Reliability Design, TSP Problems.

**Task 5:**

1. Implement and analyze run time complexity of Backtracking Application Problems: N-Queens, Sum of Subsets, Graph Coloring.

**Task 6:**

1. Implement and analyze run time complexity of Randomized Quick Sort and

**Task 7:**

1. Simulate the following CPU scheduling algorithms a) FCFS b) SJF

**Task 8:**

1. Simulate the following CPU scheduling algorithms a) Priority b) Round robin

**Task 9:**

1. Simulate Bankers Algorithm for Deadlock Avoidance
2. Implementation of Deadlock Detection Algorithm

**Task 10:**

1. Implementation of the following Memory Allocation Methods for fixed partition
   1. First Fit b) Worst Fit c) Best Fit

**Task 11:**

1. Implementation of the following Page Replacement Algorithms a) FIFO b) LRU
2. Simulate frame allocation Methods.
   1. minimum Number of frames b) equal allocation c) proportional allocation

**Task 12:**

1. Simulate the following File Organization Techniques
   1. Single level directory b) Two level Directory

**Task 13:**

1. Simulate Disk scheduling algorithms
   1. FCFS b) SSTF c) SCAN d) C-SCAN e) LOOK

**Task 14:**

1. Simulate all file allocation strategies.
   1. Sequential b) Indexed c) Linked

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Analyze running time complexity of non-recursive and recursive algorithm using asymptotic analysis. |
| CO 2 : | Implement the problems using Divide and Conquer, Greedy methods, Dynamic Programming, Backtracking, Graph Problems and analyze their time complexity. |
| CO 3 : | Simulate CPU scheduling algorithms. |
| CO 4 : | Understand memory management and virtual memory techniques and analyze various disk scheduling algorithms. |
| CO 5 : | Understand file management techniques. |

**References:**

1. Fundamental of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, Second Edition, Galgotia Publications, 2010.
2. Introduction to Algorithms, Thomas H.Cormen, Charles E.Lieserson, Ronald L.Rivest and Clifford Stein, 2nd Edition, PHI/Pearson Education, 2001.
3. The Design and Analysis of Computer Algorithms, A. Aho, J. Hopcroft and J. Ullman, Pearson Education, 1974.
4. Operating Systems: A Modern Perspective, Gary J. Nutt, Pearson Education, 1997.
5. UNIX: Concepts and Applications ,Sumitabha Das, 4th Edition, TMH, 2008
6. Operating System Concepts, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 7th Edition, John Wiley, 2005

**Online resources:**

1. http://nptel.ac.in/courses/106101060/
2. https://www.cmi.ac.in/~madhavan/teaching.html

Course Code: 22CB283

**SOFTWARE ENGINEERING LABORATORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods/week | Continuous Internal Evaluation | : | 40 Marks |
| Tutorial | : | - | Semester End Examination | : | 60 Marks |
| Credits | : | 1 | Semester End Exam Duration | : | 3 Hours |

**Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To understand the role of formal specifications in project design and be able to develop such specifications |
| 2 : | To be able to design an interface and develop a prototype for a complex software system |
| 3 : | To understand the role of testing in the software development cycle and be capable of developing a test plan |
| 4 : | To be aware of and able to use Computer Aided Software Engineering (CASE) tool |

**LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:**

**Task 1:**

Student can pick a case study for the completion of the following Tasks. (Ex : ATM, Online Railway Reservation System, etc.)

Phases in software development project, overview and need. Understand problems in existing systems and perform system analysis: Requirement analysis, SRS

**Task 2:**

Requirement analysis, SRS of both case studies

**Task 3:**

To perform the function oriented design: Data flow diagrams

**Task 4:**

To perform the function oriented design: Structured chart

**Task 5:**

To perform the user’s view analysis: Use case diagram

**Task 6:**

To draw the structural view diagram: Class diagram, object diagram.

**Task 7:**

To draw the behavioral view diagram: Sequence diagram, Collaboration diagram

**Task 8:**

To draw the behavioral view diagram: State-chart diagram

**Task 9:**

To draw the behavioral view diagram: Activity diagram

**Task 10:**

To draw the implementation view diagram: Component diagram.

**Task 11:**

To draw the implementation view diagram: deployment diagram

**Task 12:**

To perform various techniques for testing using the testing tool: Unit testing, integration.

**Task 13 :**

**Version Control System, GIT**

1. Working Locally with GIT

2. Working Remotely with GITHUB

3. Branching and Merging

4. Resolve merge Conflict

5. GIT reset and Stash operation

6. How to setup Git on Premises Hardware

7. Use Case In Devops Environment

**Task 14:**

**Continuous Integration using Jenkins**

1. Introduction of Jenkins

2. Install and setup Jenkins

3. Introduction About Maven project

4. Setup Jenkins with Maven Project

5. Continuous Build and Deployment

**Task 15:**

Identify all the problems and propose potential solutions for any of the case studies mentioned above.

**Course Outcomes:** At the end of the course, the student should be able to

|  |  |
| --- | --- |
| CO 1 : | Develop requirement specifications for a software problem in hand |
| CO 2 : | Perform functional oriented and object-oriented design |
| CO 3 : | Implement the concepts of object oriented to develop a real-world application |
| CO 4 : | Prepare test cases to rigorously test the application for ensuring quality. |
| CO 5 : | Integrate developed code using Jenkins to simulate a CI/CD pipeline |

**References:**

1. Software Engineering, Sommerville, 10th edition, Pearson Education, 2017
2. Software Engineering, A practitioner’s Approach, Roger S. Pressman, 8th edition, McGraw-Hill International Edition, 2014.
3. The Unified Development Process, Ivar Jacobson, Grady Booch, James Rumbaugh, 1st Edition, Pearson Education, 2002
4. Design Patterns: Elements of Object-Oriented Reusable Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides , Addison Wesley, 2009.
5. Software Metrics: A Rigorous and Practical Approach, Norman E. Fenton, Shari Lawrence, Pfleeger, CRC 3rd Edition, Press, 2014.
6. Fundamentals of Software Engineering, Carlo Ghezzi, Jazayeri Mehdi, Mandrioli Dino, Second Edition, Pearson Education, 2007.
7. Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices, Michael Jackson, Addison-Wesley Professional, 1995

Course Code: 22HS253/22HS352/22HS302

**INTELLECTUAL PROPERTY RIGHTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instruction | : | 3 Periods /week | Continuous Internal Evaluation | : | 100 Marks |
| Tutorial | : | - | Semester End Examination | : | - |
| Credits | : | 0 | Semester End Exam Duration | : | - |

# **Course Objectives:**

|  |  |
| --- | --- |
| 1 : | To impart basic concepts in IPR. |
| 2 : | To understand various aspects of Trademarks |
| 3 : | To create awareness on Law of Patents and Copyrights |
| 4 : | To highlight relevance of Trade Secrets in any Trade/Business |
| 5 : | To get elementary knowledge of International IPRs and New Developments in IPR |

**Unit I – Introduction to Intellectual property**

Introduction, types of intellectual property, internationalOrganizations, agencies and treaties, importance of intellectual property rights.

**Unit II– Trademarks**

Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

**Unit III – Law of copy rights**

Fundamental of copy right law, originality of material, rights of reproduction, rights to perform the work publicly, copy right ownership issues, copy right registration, notice of copy right, International copy right law.

Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

**Unit IV –Trade Secrets**

Trade secrete law, determination of trade secrete status, liability for misappropriations of trade secrets, protection for submission, trade secrete litigation.

Unfair competition: Misappropriation right of publicity, false advertising.

**Unit V – New development in intellectual property**

New developments in trade mark law; copy right law, patent law, intellectual property audits.International overview on intellectual property, international – trade mark law, copy right law,International patent law and international development in trade secrets law.

# **Course Outcomes**: At the end of the course students are able to

CO 1: Understand concepts of intellectual property rights.

CO 2: Evaluate trademark

CO 3: File for a patent

CO 4: Analyze the fairness in a competition

CO 5: Understand laws related to intellectual property rights

**Textbooks:**

1. Intellectual Property right, Deborah. E. Bo choux, 4th Edition, Cengage learning, 2012
2. Intellectual Property right – Unleashing the knowledge economy, Prabuddha Ganguli, 1st Edition, Tata McGraw Hill Publishing company ltd, 2017.

**References:**

1. Intellectual Property Patents, Trademarks and Copyrights, Richard Stim, 2nd Edition, Cengage learning, 2012.
2. Intellectual Property Rights under WTO, T. Ramappa, S. Chand, 2008.