***Course Content & Planning***

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMESTER – III** | | | |
| **Course Name** | **: Object Oriented Programming with Java** | **Course Code :** | **21CS33** |
| **Number of Lecture Hours / Week** | **: 03** | **CIE Marks :** | **50** |
| **Number of Tutorial / Practical Hours / Week** | **: 02** | **SEE Marks :** | **50** |
| **Total Number of Lecture + Tutorial/Practical Hours** | **: 40+24=64** | **SEE Duration :** | **03 Hours** |
| **L:T:P** | **: 3:0:2** | **CREDITS :** | **04** |
| **COURSE OVERVIEW**  This course is paced to provide a Basic fundamental of Java, also provides intensive hands-on experience to the students. Besides learning the basic structure and syntax of the language, students will also learn object-oriented principles and how they are applied in Java applications. | | | |
| **COURSE LEARNING OBJECTIVES (CLO)**   * Familiarize with Java environment and other integrated environment * To understand the fundamental features of object-oriented language with JAVA * To know multi-threaded programs and exception handling mechanisms * To learn Database connectivity using JDBC | | | |
| **MODULES** | | | **TEACHING HOURS** |
| **MODULE 1**  **The Java Language:** Java’s magic: the Byte code; Object-oriented programming; Simple Java program, Data types, variables and arrays, Operators, Control Statements.  **SLT**: Java Buzzwords  **Textbook1: Ch: 1, 2, 3, 4, 5** | | | **08** |
| **MODULE 2**  **Introducing Classes:** Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, This Keyword, The finalize() method, Overloading methods, Understanding static, inheritance basics, using super, creating multi-level hierarchy , Method Overriding, dynamic method dispatch, using abstract classes, using final with inheritance.  **SLT**: Garbage Collection  **Textbook1**: **Ch: 6,7, 8** | | | **08** |
| **MODULE 3**  **Packages and Interfaces:** Packages, Access Protection, Importing Packages, Interfaces, **Exception handling:** Exception handling fundamentals, Exception types, uncaught exceptions, Using Try and catch, Multiple catch clauses, Nested try statements, throw, throws, finally.  **SLT:** variables in interfaces.  **Textbook1: Ch: 9, 10** | | | **08** |
| **MODULE 4**  **Multi-Threaded Programming**: Java Thread Model, The main thread, Creating a thread, Creating multiple threads, Using isAlive() and join(), Thread priorities.  **SLT:** Thread priorities  **Textbook1: Ch: 11** | | | **08** |
| **MODULE 5**  **Java Database Connectivity:** The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with  the Database; Statement Objects; Result Set; Transaction Processing; Metadata, Data types.  **SLT**: Exceptions  **Textbook2: Ch: 6** | | | **08** |
| **PRACTICAL MODULE**   1. Develop a Java program to implement the Stack using arrays. Write Push(), Pop(), and Display() methods to demonstrate its working. 2. Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student. 3. Create a class Book which contains four members: name, author, price, num of\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects. 4. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape. 5. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods to achieve the following tasks:   • Accept deposit from customer and update the balance.  • Display the balance.  • Compute and deposit interest  • Permit withdrawal and update the balance  • Check for the minimum balance, impose penalty if necessary and update the balance.   1. Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses. 2. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrong Age( ) when the input age=father’s age. 3. Write a program which creates two threads, one thread displaying “Vidyavardhaka College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.   **Open Ended Programs**   1. Create a class building that has the public member floors, area, and occupants and a method areaperperson()respectively that display the area per person for building. In the main() method create two instance of building called house and office and display the area per person by division of area/occupants. 2. Write a class Car that contains the following attributes: The name of car The direction of car (E, W, N, S) The position of car (from imaginary zero point) The class has the following methods:   i. A constructor to initialize the attributes  ii. Turn() method to change the direction of car to one step right side (i.e. if the direction is to E, it should be changed to S and so on)  iii. Overload the turn() method to change the direction to any side directly. It should accept the direction as parameter.  iv. Move() method to change the position of car away from zero point. It should accept the distance as parameter.   1. Create a new class SalariedEmployee that extends from class Employee and overrides the method earnings( ) of class Employee to calculate the fixed monthly salary of each employee. Create object of SalariedEmployee class in EmployeeTest class (i.e, main class) to set the record for new employee and print it using toString() method. 2. Create a SavingsAccount class and use a static data member to contain the annualInterest for each of the savors. Each member of the class contains a private data member saving balance indicating the amount the saver currently has deposited. Provide a monthlyInterest() member function that calculate the monthly interest by multiplying the balance by annualInterest by 12; this interest should be added to the savingbalance. Provide a static member function modifyInterest() that sets the static annualInerest to a new value. 3. Write a program that meets the following requirements:   i. Create an array with 100 randomly chosen integers.  ii. Prompts the user to enter the index of the array, and then displays the corresponding element value. If the specified index is out of bounds, display the message Out of Bounds.  **Experiment Weightage**   |  |  |  | | --- | --- | --- | | **Type of Experiment** | **Program-No** | **Weightage** | | **Demonstration** | 6, 7, 8, | 35% | | **Exercise** | 3, 4, 5 | 35% | | **Structured Enquiry** | 1, 2 | 20% | | **Open ended** | - | 10% | | | | **24** |
| **Textbooks**   1. Herbert Schildt, “JAVA the Complete Reference”, 10th Edition, Oracle Press, 2017 2. Jim Keogh, “J2EE-TheCompleteReference”, McGraw Hill, 2017 | | | |
| **Reference Books**   1. E Balaguruswamy, “Programming with Java”, 5th Edition, McGraw Hill Education, 2017 2. R Nageswara Rao, “Core and Advanced Java, Black Book, Dreamtech Press, 2018 2. R Nageswara Rao, “Core and Advanced Java, Black Book, Dreamtech Press, 2018 | | | |
| **COURSE OUTCOMES (COs)**  At the end of the course students will be able to | | | |

|  |  |
| --- | --- |
| **CO1** | **Explain** the object-oriented concepts in JAVA |
| **CO2** | **Apply** the concepts of object-oriented programming to solve a real-world problem |
| **CO3** | **Design** data centric applications |
| **CO4** | **Design and Develop** modularized solution for the given problem statements using any IT tools**. (PO5)** |

**CO – PO – PSO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO2** | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **CO3** |  | 2 |  |  |  |  |  |  |  |  |  |  |  | 2 |  |
| **CO4** |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |
| **Avg.** | **1.5** | **2** |  |  | **3** |  |  |  |  |  |  |  |  | **2** |  |

**hello**