Object Oriented Programming Lab

CourseObjectives:

- To model a object oriented programming using abstract data types, encapsulation, inheritance and polymorphism
- Practical exposure in Fundamental features of an object oriented language like Java: object classes and interfaces, exceptions and libraries of object collections
- How to take the statement of a business problem and from this determine suitable logic for solving the problem; then be able to proceed to code that logic as a program written in Java.
- How to test, document and prepare a professional looking package for each business project using javadoc.

Detailed Contents:

Week-I

1. Write a Java program print "Hello World"
2. Write a Java program that prints all real and imaginary solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula
3. Write a Java program to implement calculator operations
4. Write a java program to find prime factors of given number
5. Write a java program to find whether given number is Palindrome or not
6. Write an application that declares 5 integers, determines and prints the largest and smallest in the group.

Week-II

1. Write a Java program to sort given list of numbers.
2. Write a Java program to implement linear search.
3. Write a Java program to implement binary search.
4. Write a java program to add two given matrices.
5. Write a java program to multiply two given matrices.
6. Write a java program for sorting a given list of names.
7. Write a Java program to give an example for command line arguments.

Week-III

1. Write a program to display details of the required employee based on his Id. The details of employee includes, Emp_name, Emp_age, Emp_gender, Emp_designation, Emp_salary, Emp_Address etc.,

2. A mail-order house sells five products whose retail prices are as follows : Product 1 : Rs. 99.90 , Product 2 : Rs. 20.20 , Product 3 : Rs. 6.87 , Product 4 : Rs. 45.50 and Product 5 : Rs. 40.49 . Each product has Prdouct_Id, Product_Name, Product_Quantity, Product_Price. Write an application that reads a series of pairs of numbers as follows :

a) product Id

b) quantity sold your program use a switch statement to determine the retail price for each product. it should calculate and display the total retail value of all products sold.

3. Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read display the complete set of unique values input after the user enters each new value.

4. Write a java program : rolling a pair of dices 10 times [each attempt should be delayed by 10000 ms] and count number Successful attempts. successful attempt : If the pair of Dice results in same values.
5. Implement the following case study using OOP concepts in Java. E-Book stall : Every book has Properties which includes : Book _Name, Book_Author, Book_Count ; Every Customer is having properties as : Customer_Id, Customer_Name, Customer_Address and he can buy Books from E-Book stall. Write a Program which will display the text book name and the remaining count of text books when a customer buys a text book.

Week-IV

1. Write an application that uses String method compareTo to compare two strings defined by the user.
2. Write an application that uses String method equals and equalsIgnoreCase to tests any two string objects for equality.
3. Write an application that uses String method indexOf to determine the total number of occurrences of any given alphabet in a defined text.
4. Write an application that uses String method concat to concatenate two defined strings.
5. Write a Java program to print all vowels in given string and count number of vowels and consonants present in given string
6. Write an application that finds the length of a given string.
7. Write an application that uses String method charAt to reverse the string.
8. Write an application that finds the substring from any given string using substring method and startsWith & endsWith methods.
9. Write an application that changes any given string with uppercase letters, displays it, changes it back to lowercase letters and displays it.

Week-V

1. Write a Java Program to implement Wrapper classes and their methods.
2. Write an application that prompts the user for the radius of a circle and uses a method called circleArea to calculate the area of the circle and uses a method circlePerimeter to calculate the perimeter of the circle.
3. Write a JAVA program for the following a. Call by value b. Call by object
4. Create a class Account with an instance variable balance (double). It should contain a constructor that initializes the balance, ensure that the initial balance is greater than 0.0. Acct details: Acct_Name, Acct_acctno, Acct_Bal, Acct_Address.
Create two methods namely credit and debit, getBalance. The Credit adds the amount (passed as parameter) to balance and does not return any data. Debit method withdraws money from an Account. GetBalance displays the amount. Ensure that the debit amount does not exceed the Account's balance. In that case the balance should be left unchanged and the method should print a message indicating "Debit amount exceeded account balance".
5. Write Java program for the following
a. Example for this operator and the use of this keyword.
b. Example for super keyword.
c. Example for static variables and methods.

Week-VI

1. Write a Java program to find Area and Circle of different shapes using polymorphism concept
2. Write a Java program which can give example of Method overloading and overriding
3. Write an application to create a super class Employee with information first name & last name and methods getFirstName(), getLastName() derive the sub-classes ContractEmployee and RegularEmployee with the information about department, designation & method displayFullName() , getDepartment(), getDesig() to print the salary and to set department name & designation of the corresponding sub-class objects respectively.
4. Derive sub-classes of ContractEmployee namely HourlyEmployee & WeeklyEmployee with information number of hours & wages per hour, number of weeks & wages per week respectively & method calculateWages() to calculate their monthly salary. Also override getDesig () method depending on the type of contract employee.
5. Write an application to create a super class Vehicle with information vehicle number,insurance number,color and methods getConsumption() displayConsumption(). Derive the sub-classes TwoWheeler and FourWheeler with method maintenance() and average() to print the maintenance And average of vehicle.
6. Extend the above TwoWheeler class with methods getType() and getName() which gives the information about the type and the name of the company.Create sub-classes Geared and NonGeared with method average() to print the average of a geared and non-geared two wheeler.

Week-VII

1. Create an abstract class Shape which calculate the area and volume of 2-d and 3-d shapes with methods getArea() and getVolume(). Reuse this class to calculate the area and volume of square ,circle ,cube and sphere.
2. Create an abstract class Employee with methods getAmount() which displays the amount paid to employee. Reuse this class to calculate the amount to be paid to WeeklyEmployeed and HourlyEmployee according to no. of hours and total hours for HourlyEmployee and no. of weeks and total weeks for WeeklyEmployee.
3. Create an Interface payable with method getAmount ().Calculate the amount to be paid to Invoice and Employee by implementing Interface.
4. Create an Interface Vehicle with method getColor(),getNumber(), getConsumption() calculate the fuel consumed, name and color for TwoWheeler and Four Wheeler By implementing interface Vehicle.
5. Create an Interface Fare with method getAmount() to get the amount paid for fare of travelling. Calculate the fare paid by bus and train implementing interface Fare.
6. Create an Interface StudentFee with method getAmount(),getFirstName(),getLastName(), getAddress(), getContact(). Calculate the amount paid by the Hostler and NonHostler student by implementing interface Student Fee

Week-VIII

1. Write a Program to create your own package. Package should have more than two classes. write a Program that uses the classes from the package.
2. Create a package named org.shapes. Create some classes in the package representing some common geometric shapes like Square, Triangle, Circle and so on. write a Program that uses the classes from the package.

3. Write a Java program to create package called dept. Create four classes as CSE, ECE, ME and CE add methods in each class which can display subject names of your respect year. access this package classes from main class
4. Write a Calculator program : Include all calculator operations in as classes in a Package "Calculator" and import in to main class5. Write a program for the following
a. Example to use interfaces in Packages. b. Example to create sub package in a package.

Week-IX

1. Program for demonstrating the use of throw, throws & finally - Create a class with a main( ) that throws an object of class Exception inside a try block. Give the constructor for Exception a String argument. Catch the exception inside a catch clause and print the String argument. Add a finally clause and print a message to prove you were there.
2. Write a program that shows that the order of the catch blocks is important. If you try to catch a superclass exception type before a subclass type, the compiler should generate errors.
3. Write a program to rethrow an exception – Define methods one() & two(). Method two() should initially throw an exception. Method one() should call two(), catch the exception and rethrow it Call one() from main() and catch the rethrown
4. Exception Handling program for ClassNotFoundException--thrown if a program can not find a class it depends at runtime (i.e., the class's ".class" file cannot be found or was removed from the CLASSPATH).
5. Exception Handling program for NumberFormatException--thrown if a program is attempting to convert a string to a numerical datatype, and the string contains inappropriate characters (i.e. 'z' or 'Q').
6. Create your own exception class using the extends keyword. Write a constructor for this class that takes a String argument and stores it inside the object with a String reference. Write a method that prints out the stored String. Create a try- catch clause to exercise your new exception.

Week-X

1. Write a program to create MyThread class with run() method and then attach a thread to this MyThread class object.
2. Write a program where the consumr thread checks the data production status [is over or not] for every 10 ms.
3. Write a Program using Threads to simulate a traffic light. The Signal lights should glow after each 10 second, one by one. For example: Firstly Red, then after 10 seconds, red will be put to off and yellow will start glowing and then accordingly green.
4. Write a Program using Threads for the following case study: Movie Theatre To watch a movie the following process is to be followed, at first get the ticket then show the ticket. Assume that N persons are trying to enter the Theatre hall all at once, display their seuence of entry into theater. Note: The person should enter only after getting a ticket and showing it to the boy.
5. Write a Program using Threads for the following case study: Train Reservation system To reserve a berth the following process need to be followed, at first check the number of available berths with the requested berths, if the number of requested berths are less than or equal to availabe berths then allot berth and print ticket or else display no berths are available. Assume that N persons are trying to reserve the berth, display their sequence of reservation status along with the number of available berths. Note : The person can print ticket only if berth is confirmed.

Week- XI

1. Write a program for the following a. display a frame with title MyFrame b. draw a horizontal line. c. Draw one line perpendicular to other. One line parallel to other.
2. Create an application to display a circle within rectangle and fill different colors in the circle & rectangle
3. Write an application that displays any string. Choose color from combo box to change the color of this displayed string and choose its size & type respectively from another two combo boxes.
4. Write a small application with a default date 01/01/2000 and three combo boxes displaying valid days, months & year (1990 – 2050). Change the displayed date with the one chosen by user from these combo boxes.
5. Create a GUI with title STUDENT which has labels roll no., name, course, gender, class, address with textboxes for taking input from the user(without any functionality) and checkboxes for selecting the course, radio buttons for selecting gender with appropriate background color.
6. Create a GUI application to display a calculator using grid Layout (You do not have to provide functionality).

Week-XII

1. Write a program to create a frame by creating an object to Jframe class and include close button to terminate the application of the frame.
2. Write program for the following.

a. Display text in the frame by overriding PaintComponent() method of Jpanel class.
b. Display some text in the frame with the help of a Label.

3. Write a program to create a push button , when the button is clicked an image is displayed in the frame.
4. Write a program to create a menu with several menu items.
5. Create an application Form for University Enroll ment with the following Fields.

a. Check box b. Text area c. List box d. Display text e. Push buttons f. Combo box. g. Radio buttons. h. Back ground color

Week-XIII

1. Write a program to insert data into Student Table.
2. Write a program to retrieve the data from the table Student.
3. Create a Form to insert and retrieve the data from Database as user prefer.
4. Write a program to store an Image and retrieve an image from Database
5. Write a program to Store and retrieve file content from the Data base.

Course Outcomes:

CO 1: Be able to analyze and design a computer program to solve real world problems based on object-oriented principles.
CO 2: Be able to write simple GUI interfaces for a computer program to interact with users, and to understand the event-based GUI handling principles.
CO 3: A competence to design, write, compile, test and execute straightforward programs using a high level language.
CO 4: Demonstrate the ability to employ various types of selection constructs in a Java program. Be able to employ a hierarchy of Java classes to provide a solution to a given set of

requirements.

CO 5: Become familiar with the fundamentals and to acquire programming skills in the Java language.
