Gujarat Technological University

Sarvajanik College of Engineering & Technology

B.E. –III, SEM - V (Computer Engineering)

Subject Name: Object Oriented Programming using Java(2150704)

Unit 1: Basics of Java

1. Write a program that calculate percentage marks of the student for six subjects and display grade.
2. Write a program to enter two numbers and perform mathematical operations on them.

Unit 2: Array and String

1. Write a simple java application that sorts the integer numbers passed through command line.
2. Write a program to find length of string and print second half of the string.
3. Write a program to accept a line and check how many consonants and vowels are there in line.
4. Write a program to count the number of words that start with capital letters.
5. Write a program to find that given number or string is palindrome or not.
6. Create a class which asks the user to enter a sentence, and it should display count of each vowel type in the sentence. The program should continue till user enters a word “quit”. Display the total count of each vowel for all sentences.

Unit 3: Classes, Objects and Methods

1. Write a simple java application that defines a class Student with roll_no(int), name(String), address(String) & branch(String) as data fields. The class should have getData() & showData() methods. The program should create an array of Student object, get the details and display it.

2. Write a simple java application that defines a class Complex with real(int) and img(int) as data fields, no-argument constructor and parameterized constructor. The class should have overloaded methods to perform addition of two Complex numbers by passing objects as arguments. Demonstrate this keyword in parameterized constructor.

Unit 4: Inheritance and Interface

1. Write a simple java application that creates a Player class. Inherit CricketPlayer class from Player class. Inherit Batsman & Bowler classes from CricketPlayer class. Assume suitable data and member methods.

2. Write a simple java application that creates a Shape class with two double data members. The class should have area methods to calculate the area of shape. Inherit two classes Rectangle and Triangle from Shape class. Demonstrate method overriding & super keyword.

3. Write a simple java application that creates a Shape class with two double data members. The class should have one abstract method area to calculate the area of shape. Inherit two classes Rectangle and Triangle from Shape class. Demonstrate run time polymorphism.

4. Write a simple java application that creates an interface Shape. The interface declares read() and show() methods and PI as constant data member. Create classes Rectangle and Circle that implement shape interface. Assume suitable data and member methods.

Unit 5: Package

1. Create two packages, CE_Dept & IT_Dept, both with Machine_Detail as class. The class should have a method to display machine information (No_of_PC(int), configuration(String)) for both departments. Write a java application that imports both defined packages and call their methods.

Unit 6: Exception Handling

1. Write a simple java application that reads marks of five subjects through command line arguments and display average. The application should generate an exception if marks are not in integer format and out of 0-100.
2. Write a java program using nested try-catch blocks. If user enters only one command line argument than inner try block should throw an Exception. If user enters two command line arguments, divide first argument by second argument. If second argument is zero than proper exception should be handled.
3. Write a simple java application that declares Employee class. The program should generate and handle custom exceptions such as
    a. InvalidEmailAddressException if the address does not contains . and @
    b. InvalidTelephoneNumberException if total no of digits > 10.

Unit 7: Multithreaded Programming

1. Write a simple java application that creates two threads: One thread creates even numbers and another thread creates odd numbers.
2. Implement producer consumer IPC problem using multi-threading.

Unit 8: IO Programming

1. Create a class called Student. Write a student manager program to manipulate the student information from files by using FileInputStream and FileOutputStream.

Unit 10: Networking with java.net

1. Study of Network Programming. Write a simple client-server java program. The client sends radius of circle to the server. The server replies back with area of circle.