Maulana Abul Kalam Azad University of Technology, West Bengal

Laboratory Examination, Odd Semester,2021-2022

Subject: OOP Lab **Code: PCC-CS593**
Semester: 5th **Time: 3 hours**
Full Marks: 60

Marks Distribution:

[**Problem Definition :10 Program: 20 Output: 10 Viva: 20**]

Q1.

1.1. Write a program for multiple catch to fire ArrayIndexOutOfBoundsException and StringIndexOutOfBoundsException both.

1.2. Show that the String class type objects are immutable but StringBuffer class objects are mutable.

Q2.

2.1. Write a Java Program to demonstrate Dynamic Method Dispatch.

2.2. Create a class Employee is having instance variables name and id. Create its subclass named Scientist which has instance variables no_of_publication and experience. Now create its subclass, say DScientist which has instance variable award. Put a method like: public String toString(){} in every class where you describe about the class and from main() method create object of each class and print each object.

Q3.

3.1. Create a class Parent having non-static variables id, name and address. Create a class ChildOne having non-static variables id, name, address and marks. Also create another class ChildTwo with non-static variables id, name, address, qualification and salary. Design the program and use object of each class from main().

3.2. Write a program to demonstrate anonymous inner class.

Q4.

4.1. Write a program to take a sentence and convert it into string arrays and sort the words using any Sorting technique.

4.2. Show that static variable of a class only has one copy for different object but instance variable may have separate copy for individual object.

Q5.

5.1. Write a program in Java which will read a string and rewrite it in the alphabetical order. For example, the word JAVA should be written as AAJV.

5.2.
Create a class with a method void show () and make 3 subclasses of it and all subclasses have void show () method overridden and call those methods using their class references.

Q6.

6.1. Show that static block is executed at the time of class loading and also the order of execution of these blocks (for multiple blocks/ inherited block).

6.2. Write a program to calculate GCD of two numbers where numbers you have to take as user input.

Q7.

7.1. Create a base class Fruit with name, taste and size as its attributes. Create a method called eat() which describes the name of the fruit and its taste. Inherit the same in 2 other classes Apple, and Orange and override the eat() method to represent each fruit taste.

7.2 Take a sting from keyboard and convert into character array (new one).

Q.8

8.1. Write a Java program to show that non-static block(ordinary block/instance block)is executed when object is created and also the order of execution of these blocks(for multiple blocks/inherited blocks).

8.2. Write a program to find sum and average of several integers (in an array) using enhanced-for loop.

Q9.

9.1. Create a class named Animal which includes methods like eat() and sleep().Create a child class of Animal named ‘Bird’ and override the parent class methods. Add a new method named fly().

Create an instance of Animal class and invoke the eat and sleep methods using that object.

Create an instance of Bird class and invoke the eat, sleep and fly methods using this object.

9.2. Write a program to handle the ArithmeticException.

Q10.

10.1. Write a program to find the sum of digits of a given integer number (take user input).

10.2.
Write a program to fire any checked exception manually using ‘throw’ keyword.

Q11.

11.1. Create a class named ‘box’ that include integer data fields for length, width and height. Create three constructors that require one, two and three arguments respectively.

When one argument is used assign it to length, assign zeroes to height and width and print “Line Created”. When two arguments are used to assign them to length and width, assign zero to height and print “Rectangle Created”. When three arguments are used assign them to the three variables and print “Box Created”.

11.2. Write a Java program to achieve Multithreading by implementing Runnable interface.

Q12.

12.1.

Assume that a bank maintains two kinds of account for its customers, one called savings account and other called 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 (say Rs. 1000) and if the balance falls below this level a service charge is imposed (say Rs. 100).
Create a class **Account** that stores customer name, account number and type of account. From this class derive two classes **Curr_Acct** and **Savn_Acct** respectively to make them more specific to their requirements. Include the necessary methods to achieve the following tasks:
a. Accept deposit from a customer and update the balance.
b. Display the balance.
c. Compute and deposit interest.
d. Permit withdrawal and update the balance.
e. Check for minimum balance, impose penalty, if necessary, and update the balance.

Use constructors to initialize the class members.

12.2. Write a Java program to find the length of the string taken from keyboard and also find the length of the string except front and end spaces.

Q13.

13.1.

Find length of a string taken from keyboard and also find the length of that string except front and end spaces.

13.2. Write a program to create a user defined exception named PayOutOfBoundsException (provided the monthly salary of a person is less than Rs. 10,000 /-) and fire the exception.

Q14.

14.1. Generate password from initials of one's first_name, middle_name, last_name and with last four digit of your roll_no.(if middle name not presents, it won't come)

14.2.
Define an object reference and initialize it to null. Try to call a method through this reference. Now wrap the code in a try-catch clause to catch the exception.

Q15.

15.1. Create an interface named Shape with a field pie (=3.14). Create two subclasses of it named Circle and Rectangle create object of the two classes and calculate their area.

15.2. Create a class and test if method overloading holds good for return type of method or not.

Q16.

16.1. Write a program to show that finally block will execute even exceptions are not handled.

16.2. Create two **interface**s, each with single method. Inherit a new **interface** from the two, adding a new method. Create a class by implementing the new **interfaces** and also inheriting from an abstract class (with one abstract method). In **main( )**, create an object of your class and call all the implemented methods.

Q17.

17.1. Write a Java program to demonstrate Method Hiding.

17.2. Write a Java Program to demonstrate multiple inheritance can be achieved in Java in interface level.

Q18.

18.1. Write a Java program to show that non-static block(ordinary block/instance block)is executed when object is created and also the order of execution of these blocks(for multiple blocks/inherited blocks).

18.2. Write a Java program to implement Thread by extending Thread class. Print the name of main thread and also print the name of child thread.