

1st ASSESSMENT

OBJECT ORIENTED TECHNIQUES
USING JAVA[BCSE0352]

Total Marks:30

Section A Quiz:10X1=10

Section B Marks :4X5=20

Date:20/09/2024

Notes: Attempt Any four questions, all carries equal marks

* After completing all questions in Section B,
Open your MS Office and write your:

- a) Name
- b) Roll No
- c) Branch
- d) Section
- e) Workshop Lab Number

Then, put all four program source codes along with their output screenshots and convert it into a PDF file named as NAME_BRANCH_SECTION.

*Upload this PDF file in MS Forms

Section B

QUESTION 1: Write a Java program that uses decision-making statements to determine and display the performance level of an employee based on their evaluation scores.:

Instructions:

- a) Prompt the user to enter the class name (use "**EmployeePerformance**" as the class name).
- b) Use the Scanner class to input scores for the following five evaluation criteria: **Work Quality, Punctuality, Teamwork, Problem-Solving, and Communication.**
- c) Calculate the overall percentage based on the scores entered.
- d) Assign a **performance level** based on the following criteria:
 - **Outstanding:** 90% and above
 - **Excellent:** 80% to 89%
 - **Good:** 70% to 79%
 - **Average:** 60% to 69%
 - **Below Average:** 50% to 59%
 - **Unsatisfactory:** Below 50%
- e) Display the calculated percentage and the corresponding performance level.

QUESTION 2: Write a Java program to generate the factorial of a given number. The program should:

Instructions:

- a) Use a class named **FactorialCalculator**.
- b) Prompt the user to enter a number to calculate its factorial using the Scanner class.
- c) Implement a user-defined method called **calculateFactorial** that takes an integer as a parameter and returns the factorial of that number.
- d) Use a **for loop** within the **calculateFactorial** method to compute the factorial.
- e) Display the factorial of the entered number.

QUESTION 3: Write a Java program to calculate the volume and surface area of different 3D shapes (Sphere and Cuboid). The program should:

Instructions:

a. Use an **abstract class** named **Solid** with the following:

- Two abstract methods: `volume()` and `surfaceArea()`, both returning a double.
- A regular method `display()` that prints "This is a solid shape."

b. Create two subclasses:

- A Sphere class that:
 - Has a constructor accepting the radius as a parameter.
 - Implements the `volume()` method to calculate the volume of the sphere.
 - Implements the `surfaceArea()` method to calculate the surface area of the sphere.
- A Cuboid class that:
 - Has a constructor accepting the width, height, and depth as parameters.
 - Implements the `volume()` method to calculate the volume of the cuboid.
 - Implements the `surfaceArea()` method to calculate the surface area of the cuboid.

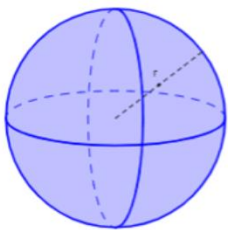
c. In the **Main** class:

- Create objects of Sphere and Cuboid using the Solid reference.
- Print the volume and surface area of both shapes using the `volume()` and `surfaceArea()` methods.

d. Demonstrate the use of constructors and polymorphism by printing the calculated **volume** and **surface area** of the shapes.

Notes:

Surface Area and Volume of Sphere



$$\text{Surface Area} = 4\pi r^2$$

$$\text{Volume} = \frac{4}{3}\pi r^3$$

Question 4: Write a Java program to manage Employee details using the concepts of default constructor, no-arg constructor defined by the programmer, and parameterized constructor. The program should:

Instructions:

a. Create a class named Employee with:

- Private fields for name (String), position (String), and salary (double).

b. Implement the following constructors in the Employee class:

- A **default constructor** that initializes name to "Unknown", position to "Unassigned", and salary to 0.0.
- A **no-arg constructor** defined by the programmer that prints a message saying "No argument constructor called."
- A **parameterized constructor** that accepts arguments for name, position, and salary, and initializes the respective fields.

c. Implement a method named **displayEmployeeDetails()** to print the employee's details (name, position, and salary).

d. In the **Main** class:

- Create three Employee objects:
 - One using the **default constructor**.
 - One using the **no-arg constructor** defined by the programmer.
 - One using the **parameterized constructor**, passing values for the name, position, and salary.

e. Display the details of each employee by calling the **displayEmployeeDetails()** method on each object.

f. Demonstrate the constructor usage:

- Explain how each constructor initializes the fields differently and observe the output.

.

Question 5: Write a Java program to manage student details and calculate their scholarship based on their academic performance and extracurricular participation. The program should use the concepts of constructors, abstract class, decision making, loops, Scanner, and methods. The program should:

Instructions:

a. Create an **abstract class** named *Student* with the following:

- Private fields for **name** (String) and **tuitionFee** (double).
- A constructor to initialize the **name** and **tuitionFee**.
- An abstract method **calculateScholarship()** that returns a double.
- A concrete method **displayDetails()** that prints the student's name and tuitionFee.

b. Create two subclasses:

- A **MeritStudent** class that:
 - Inherits from **Student**.
 - Has a constructor that accepts **name** and **tuitionFee**.
 - Implements the **calculateScholarship()** method to calculate a scholarship based on a percentage of the tuition fee (fixed for merit-based students).
- A **SportsStudent** class that:
 - Inherits from **Student**.
 - Has a constructor that accepts **name** and **tuitionFee**.
 - Implements the **calculateScholarship()** method to calculate a scholarship based on performance in both academics and sports (extracurricular rating provided by the user) along with the tuition fee.

c. In the **Main** class:

- Prompt the user using the **Scanner** class to enter details for multiple students (either **MeritStudent** or **SportsStudent**), including their name, tuitionFee, and extracurricular rating (for sports students).
- Use a loop to allow the user to enter details for more than one student.

- After creating each student object, use decision-making (if or switch) to determine if the student is a **MeritStudent** or **SportsStudent**, and then call the respective methods to calculate the scholarship.
- Display the student's details and calculated scholarship using the **displayDetails()** and **calculateScholarship()** methods.

d. Demonstrate the use of decision-making and loops to handle input for multiple students and different student types.