

## **Objects and Classes:**

### Question 1:

Create a class Book that has the following:

Fields: title, author, price

Method: displayInfo() to display the book details.

In the main method, create two Book objects, assign values to the fields, and call the displayInfo() method for both objects.

### Question 2:

Write a program that creates a class Rectangle with the following:

Fields: length and width

Constructor: Initialize length and width

Method: calculateArea() to return the area of the rectangle

In the main method, create an object of the Rectangle class, take user input for length and width, and display the area of the rectangle.

### Question 3:

Write a class Student with the following:

Fields: name, rollNumber, marks

Constructor: To initialize all fields

Method: checkPass() to check if the student passes (marks  $\geq 50$ ).

In the main method, create an array of 3 students, initialize their values using the constructor, and check if each student has passed or failed.

### Question 4:

Create a class Circle with the following:

Field: radius

Method:

setRadius() to set the value of radius

calculateCircumference() to return the circumference (use formula  $2 * \text{Math.PI} * \text{radius}$ )

In the main method, create an object of Circle, set the radius using the setRadius() method, and display the circumference.

Question 5:

Write a class Car with the following:

Fields: brand, model, year

Method: displayDetails() to display car details.

In the main method:

Create an array of Car objects with 2 cars.

Assign values to each object and call the displayDetails() method for both cars.

Question 6:

Write a class Person with the following:

Fields: name (String), age (int)

Methods:

getName() and setName() for the name field

getAge() and setAge() for the age field

In the main method, create an object of the Person class. Use the setter methods to set the values of name and age and use the getter methods to display the values.

Question 7:

Create a class Employee with the following:

Fields: employeeId (int), salary (double)

Methods:

Getter and setter for employeeId

Getter and setter for salary

In the main method, create an object of Employee, set the values of employeeId and salary, and display them using the getter methods.

Question 8:

Write a class Product with the following:

Fields: productName (String), price (double)

Methods:

Getter and setter for productName

Getter and setter for price

In the main method:

Create an array of 3 Product objects.

Use setters to set the values for each product.

Use getters to display the product details.

Question 9:

Write a class BankAccount with the following:

Fields: accountNumber (String), balance (double)

Methods:

getAccountNumber() and setAccountNumber() for accountNumber

getBalance() and setBalance() for balance

In the main method:

Create a BankAccount object.

Set the accountNumber and balance using setters.

Display the details using the getter methods.

Question 10:

Create a class Movie with the following:

Fields: title (String), rating (double), releaseYear (int)

Methods:

Getter and setter for title

Getter and setter for rating

Getter and setter for releaseYear

In the main method, create two Movie objects, set their fields using setters, and display their details using getters.

### **Inheritance:**

#### Question 1:

Create a base class Vehicle with the following:

A field speed (int).

A method displaySpeed() to print the speed.

Create a derived class Car that inherits from Vehicle and adds:

A field fuelType (String).

A method displayFuelType() to print the fuel type.

In the main method, create an object of Car, set the speed and fuel type, and display both using the methods.

#### Question 2:

Write a class Person with a field name and a method displayName().

Create a subclass Student that adds a field studentId and a method displayStudentId().

In the main method, create a Student object, set the name and student ID, and display both.

#### Question 3:

Create a base class Shape with:

A method area() that prints "Area not defined".

Create a derived class Rectangle that overrides the area() method and calculates the area using length and width fields.

In the main method, create an object of Rectangle, set the dimensions, and display the area.

Question 4:

Write a class Animal with a method sound() that prints "Animals make sound".

Create two subclasses Dog and Cat that override the sound() method to print their specific sounds.

In the main method:

Create an array of Animal references and assign Dog and Cat objects.

Use a loop to call the sound() method for each object.

Question 5:

Create a base class Employee with fields name and salary.

Create a derived class Manager that adds a field bonus.

In the main method:

Create an object of Manager.

Set the name, salary, and bonus.

Write a method to display the total compensation (salary + bonus).

Question 6:

Create a class BankAccount with fields accountNumber and balance.

Write a method displayDetails() to show the account details.

Create a subclass SavingsAccount that adds a field interestRate.

Override the displayDetails() method to include the interest rate.

In the main method, create an object of SavingsAccount and display the details.

Question 7:

Write a class Parent with a method show() that prints "This is parent class".

Create a class Child that inherits Parent and overrides the show() method to print "This is child class".

In the main method:

Create objects of Parent and Child classes.

Call the show() method using both objects.

Question 8:

Create a base class Shape with a method perimeter() that prints "Perimeter not defined".

Create a subclass Circle with a field radius.

Override the perimeter() method to calculate and print the perimeter of the circle ( $2 * \text{Math.PI} * \text{radius}$ ).

In the main method, create a Circle object, set the radius, and display the perimeter.

Question 9:

Write a class Vehicle with a field model and a method displayModel().

Create two subclasses:

Car adds a field seats (int).

Bike adds a field hasCarrier (boolean).

In the main method, create objects of Car and Bike, set their fields, and display all details.

Question 10:

Create a base class Publication with fields title and price.

Create two subclasses:

Book adds a field author.

Magazine adds a field issueNumber.

In the main method:

Create an object of Book and Magazine.

Set their fields and display the details of both objects.

## **Polymorphism:**

### Question 1

Write a program to demonstrate method overriding using a superclass Shape and subclasses Circle and Rectangle. Each subclass should override a draw() method.

### Question 2

Write a program to create a superclass Employee and two subclasses Manager and Developer. Add a method calculateSalary() in each class. Use polymorphism to calculate and display the salary of employees.

### Question 3

Create a base class Vehicle and two derived classes Car and Bike. Add a method run() in the base class and override it in the subclasses. Use an array of Vehicle to demonstrate runtime polymorphism.

### Question 4

Write a program to create a class Animal and two subclasses Dog and Cat. Use the instanceof operator to call a specific method fetch() for Dog and scratch() for Cat while iterating through an array of Animal.

### Question 5

Write a program that uses method overloading to add two integers, two doubles, and two strings. Demonstrate compile-time polymorphism.

These questions comprehensively cover key concepts of Java polymorphism and provide ample opportunities for students to write, analyze, and test their understanding of code.

## **ArrayList, searching, sorting:**

### 1. Create and Populate an ArrayList

Write a Java program to create an ArrayList of integers. Add 10 random numbers to the list and print the list.

### 2. Search for an Element

Write a program that creates an ArrayList of String objects. Check if the list contains the string "Java", and print the index if found. Use contains() and indexOf().

### 3. Remove Elements

Write a Java program to create an ArrayList of integers. Remove all even numbers from the list using an Iterator.

#### 4. Searching a Minimum Value

Write a program to find the minimum value in an ArrayList of integers using a for-loop.

#### 5. Insertion Sort Implementation

Write a program to sort an ArrayList of integers using the insertion sort algorithm. Print the sorted list.

#### 6. Find Duplicates in an ArrayList

Write a program to find and print duplicate elements in an ArrayList of integers. Use a HashSet to identify duplicates.

### **Lambdas and Streams:**

#### Question 1:

Write a lambda expression to calculate the square of a number.

Use a functional interface to implement it.

Call the lambda expression with a value of 5 and print the result.

#### Question 2:

Create a List of integers: {1, 2, 3, 4, 5}.

Use a stream to:

Multiply each element by 2.

Collect the results into a new list.

Print the new list.

#### Question 3:

Write a lambda expression to check if a string starts with a specific letter (e.g., 'A').

Use the functional interface Predicate<String>.

Test the lambda with strings "Apple" and "Banana" and print the results.



Question 4:

Given a List of names: {"Alice", "Bob", "Charlie", "Anna", "David"},

Use a stream to:

Filter the names that start with 'A'.

Print the filtered names.

Question 5:

Write a lambda expression to sort a List of integers in descending order.

Use the Collections.sort method.

Test it with a list: {5, 2, 9, 1, 6} and print the sorted list.

Question 6:

Create a List of numbers: {3, 7, 2, 8, 6, 1}.

Use a stream to:

Filter numbers greater than 5.

Find the sum of those numbers using reduce().

Print the result.

Question 7:

Write a program to create a list of String values: {"Java", "Python", "C++", "JavaScript"}.

Use streams to:

Convert all strings to uppercase.

Print the resulting list.

Question 8:

Create a list of integers {1, 2, 3, 4, 5}.

Use a stream to:

Find the product of all elements using reduce().

Print the result.

Question 9:

Write a lambda expression to find the maximum of two numbers.

Use a functional interface.

Test it with two numbers: 10 and 20, and print the maximum.

Question 10:

Create a List of names: {"Tom", "Jerry", "Spike", "Tyke"}.

Use a stream to:

Count how many names have more than 3 letters.

Print the count.