

Object Oriented Programming – Assignment for II Sem CSE A, B, C

Dead Line to Submit – April 15 to 20, 2025

Instructions to submit:

- ✓ Follow the given template to submit the Assignment.
- ✓ For every program along with the code, the errors along with the rectification of them should be mentioned.
- ✓ While displaying the output, the first print statement should be your Name, roll no, Section and followed with the output.
- ✓ Important points / Explanation should be given to each program.
- ✓ Everyone should prepare a soft copy and a hard copy (Hand written can also be submitted).
- ✓ Viva questions will be asked during the submission of the assignment, based on the performance marks will be allotted.
- ✓ The student has to show the executions of all the programs.

1. Write a java program with class named “book”. The class should contain various attributes such as “title, author, yearofpublication”. It should also contain a “constructor” with parameters which initializes “title”, ”author”, and “yearofpublication”. Create a method which displays the details of the book i.e. “author, title, yearofpublication”. (Display the details of two books i.e. create 2 objects and display their details).
2. Write a java program with class named “MyClass”, with a static variable “count” of “int” type, initialized to “0” and a constant variable “PI” of type “double” initialized to 3.14159 as attributes of that class. Now define a constructor for “MyClass” that increments the “count” variable each time an object of “MyClass” is created. Finally print the final values of “count” and “PI” variables.
3. Define a Java class named **VisibilityExample** with the following attributes and methods:

Attributes:

- A public integer variable named publicVariable, initialized to 10.
- A private integer variable named privateVariable, initialized to 20.

Methods:

- A public method named publicMethod() that prints "This is a public method."
- A private method named privateMethod() that prints "This is a private method."
- In a separate Java class named **Main**, write the main method to demonstrate accessing the members of the VisibilityExample class:
- Create an object of the VisibilityExample class.
- Access and print the value of the public variable publicVariable.
- Call the public method publicMethod().

- Attempt to access the private variable `privateVariable` and call the private method `privateMethod()` in the Main class.
- **Note:** attempting to do so will result in a compilation error.

4. Write a Java program that takes a number from the user and generates an integer between 1 and 7. It displays the weekday name (Use Conditional Statements).

Ex: Sample Input

Input number: 3

Expected Output :

Wednesday

5. Write a Java program to display the multiplication table of a given integer.

Ex: Sample Input

Input the number (Table to be calculated) : Input number of terms : 5

Expected Output :

5 X 0 = 0

5 X 1 = 5

5 X 2 = 10

5 X 3 = 15

5 X 4 = 20

5 X 5 = 25

6. Write a Java program that reads two floating-point numbers and tests whether they are the same up to three decimal places (Use Conditional Statements).

Ex: Sample Input

Input floating-point number: 25.586

Input floating-point another number: 25.589

Expected Output :

They are different

7. Write a program that accepts three numbers from the user and prints "increasing" if the numbers are in increasing order, "decreasing" if the numbers are in decreasing order, and "Neither increasing or decreasing order" otherwise (Use Conditional Statements).

Ex: Sample Output

Input first number: 1524

Input second number: 2345

Input third number: 3321

Expected Output :

Increasing order

8. Write a Java program that reads a positive integer and count the number of digits the number (less than ten billion) has (Use Conditional Statements).

Ex: Sample Output

Input an integer number less than ten billion: 125463

Expected Output :

Number of digits in the number: 6

9. Write a Java program to display Pascal's triangle.

Ex: Sample Output

Input number of rows: 5

Expected Output :

Input number of rows: 5

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

10. Write a Java program to display the following character rhombus structure.

Ex: Sample Output

Input the number: 7

Expected Output :

```
    A
   ABA
  ABCBA
 ABCDCBA
ABCD EDCBA
ABCDE F EDCBA
ABCDEF G FEDCBA
ABCDEF EDCBA
 ABCDCBA
  ABCBA
   ABA
    A
```

11. Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model,

year, and fuel type. Implement methods for calculating fuel efficiency, distance travelled, and maximum speed.

12. Write a Java program to create a class called Employee with methods called work () and getSalary(). Create a subclass called HRManager that overrides the work () method and adds a new method called addEmployee().
13. Create a calculator using the operations including addition, subtraction, multiplication and division using multi-level inheritance and display the desired output.
14. Consider a software system for a company that manages its employees. The company categorizes its employees into two primary types: RegularEmployee and Manager. Both types of employees share common attributes such as name and employee ID, but managers have attributes such as a bonus. You are tasked with designing the Java classes for this scenario and add up the salary for each type.
15. A superclass named "Shapes" has a method called "area()". Subclasses of "Shapes" can be "Triangle", "circle", "Rectangle", etc. Each subclass has its own way of calculating area. Using base class as Shapes with subclasses triangle, circle and rectangle, use overriding polymorphism and find the area for each shape.
16. creating one superclass Animal and three subclasses, Herbivores, Carnivores, and Omnivores. Subclasses extend the superclass and override its eat() method. Returning the method for the required type of animals.
17. Write a Java program to create an abstract class Animal with an abstract method called sound(). Create subclasses Lion and Tiger that extend the Animal class and implement the sound() method to make a specific sound for each animal.
18. Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.

19. What will be the output of the following program?

```
interface A
{
    void Method ();
}

class B
{
    public void Method ()
    {
        System. out.println ("My Method");
    }
}
```

```

class C extends B implements A
{

}

class Main
{
    public static void main (String [] args)
    {
        A a = new C ();

        a. Method ();
    }
}

```

19. Does below code compile successfully? If not, why?

```

interface A
{
    int i = 111;
}

class B implements A
{
    void methodB()
    {
        i = 222;
    }
}

```

20. Write a Java program to create an interface Shape with the getPerimeter() method. Create three classes Rectangle, Circle, and Triangle that implement the Shape interface. Implement the getPerimeter() method for each of the three classes.
21. Write a Java program that creates a class hierarchy for employees of a company. The base class should be Employee, with subclasses Manager, Developer, and Programmer. Each subclass should have properties such as name, address, salary, and job title. Implement methods for calculating bonuses, generating performance reports, and managing projects.
22. Write a Java program to create a class called Student with private instance variables student_id, student_name, and grades. Provide public getter and setter methods to access and modify the student_id and student_name variables. However, provide a method called addGrade() that allows adding a grade to the grades variable while performing additional validation.
23. Write a Java program to create a base class BankAccount with methods deposit() and withdraw(). Create two subclasses SavingsAccount and CheckingAccount. Override the withdraw() method in each subclass to impose different withdrawal limits and fees.

24. Write a Java program to create an abstract class Bird with abstract methods fly() and makeSound(). Create subclasses Eagle and Hawk that extend the Bird class and implement the respective methods to describe how each bird flies and makes a sound.
25. Write a Java program to create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.
26. Write a Java programming to create a banking system with three classes - Bank, Account, SavingsAccount, and CurrentAccount. The bank should have a list of accounts and methods for adding them. Accounts should be an interface with methods to deposit, withdraw, calculate interest, and view balances. SavingsAccount and CurrentAccount should implement the **Account interface** and have their own unique methods.
27. How would you demonstrate the initialization and usage of arrays in Java? Discuss the various methods of declaring, initializing, and populating arrays. Using the arrays concept write a java program to initialize a matrix, addition of two matrices, multiplication of two matrices and display the output.
28. a. Discuss the difference between the Interfaces vs. Abstract Classes in detail.
- b. Discuss the difference between the Overriding vs. Overloading in detail.
29. (Triangle class) Design a new Triangle class that extends the abstract GeometricObject class. Draw the UML diagram for the classes Triangle and GeometricObject and then implement the Triangle class. Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a Triangle object with these sides and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not.
30. Rewrite the PrintCalendar class in Listing 6.12 to display a calendar for a specified month using the Calendar and GregorianCalendar classes. Your program receives the month and year from the command line. For
- example:
java Exercise13_04 5 2016
This displays the calendar shown in Figure.

```
Command Prompt
c:\exercise>java Exercise13_04 5 2016
May, 2016
-----
Sun Mon Tue Wed Thu Fri Sat
  1  2  3  4  5  6  7
  8  9 10 11 12 13 14
 15 16 17 18 19 20 21
 22 23 24 25 26 27 28
 29 30 31

c:\exercise>
```