**NAME – Piyush Vijay Lende**

**AF CODE – AF04953355**

Java Basics and OOPs Assignment

## 1. What is Java? Explain its features.

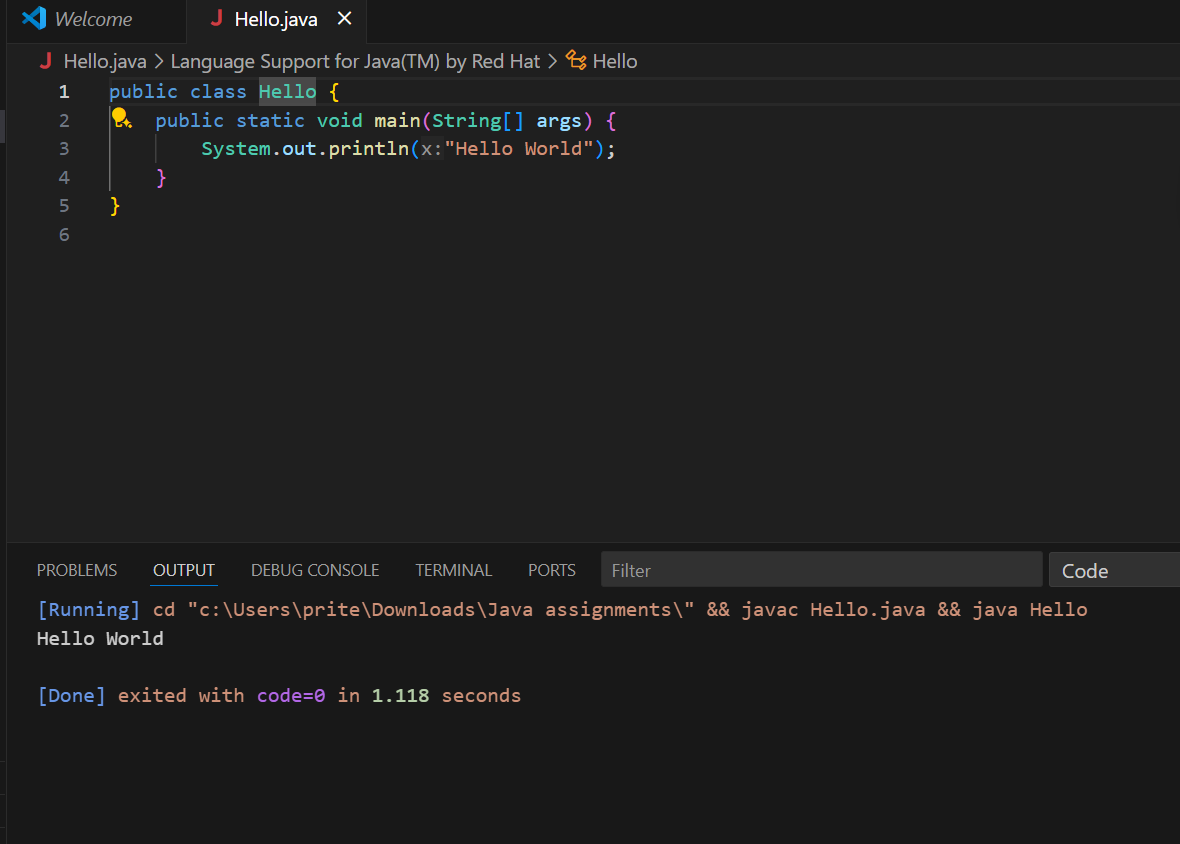
Java is a high-level, object-oriented programming language developed by Sun Microsystems (now owned by Oracle). It is platform-independent, secure, and widely used for building web and mobile applications.  
  
Features:  
- Platform Independent  
- Object-Oriented  
- Simple and Secure  
- Robust  
- Multithreaded  
- Portable  
- Distributed

## 2. Explain the Java program execution process.

1. Write code in a .java file.  
2. Compile it using javac to create a .class file (bytecode).  
3. JVM executes the bytecode on any platform.

## 3. Write a simple Java program to display 'Hello World'.

public class HelloWorld {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 }  
}



## 4. What are data types in Java? List and explain them.

Java has:  
1. Primitive types: int, float, double, char, boolean, byte, short, long  
2. Non-primitive types: String, Array, Class, Object

Example:

java

CopyEdit

int age =19;

String name = "Piyush";

## 5. What is the difference between JDK, JRE, and JVM?

JDK: Java Development Kit (tools for development)  
JRE: Java Runtime Environment (runs Java apps)  
JVM: Java Virtual Machine (executes bytecode)

## 6. What are variables in Java? Explain with examples.

Variables store data. Example:  
int age = 19;  
String name = "Piyush";

## 7. What are the different types of operators in Java?

* Arithmetic: +, -, \*, /, %
* Relational: ==, !=, >, <, >=, <=
* Logical: &&, ||, !
* Assignment: =, +=, -=, etc.
* Unary: ++, --
* Bitwise: &, |, ^

## 8. Explain control statements in Java (if, if-else, switch).

**1. if Statement**

Executes a block of code only if a specified condition is true.

**2. if-else Statement**

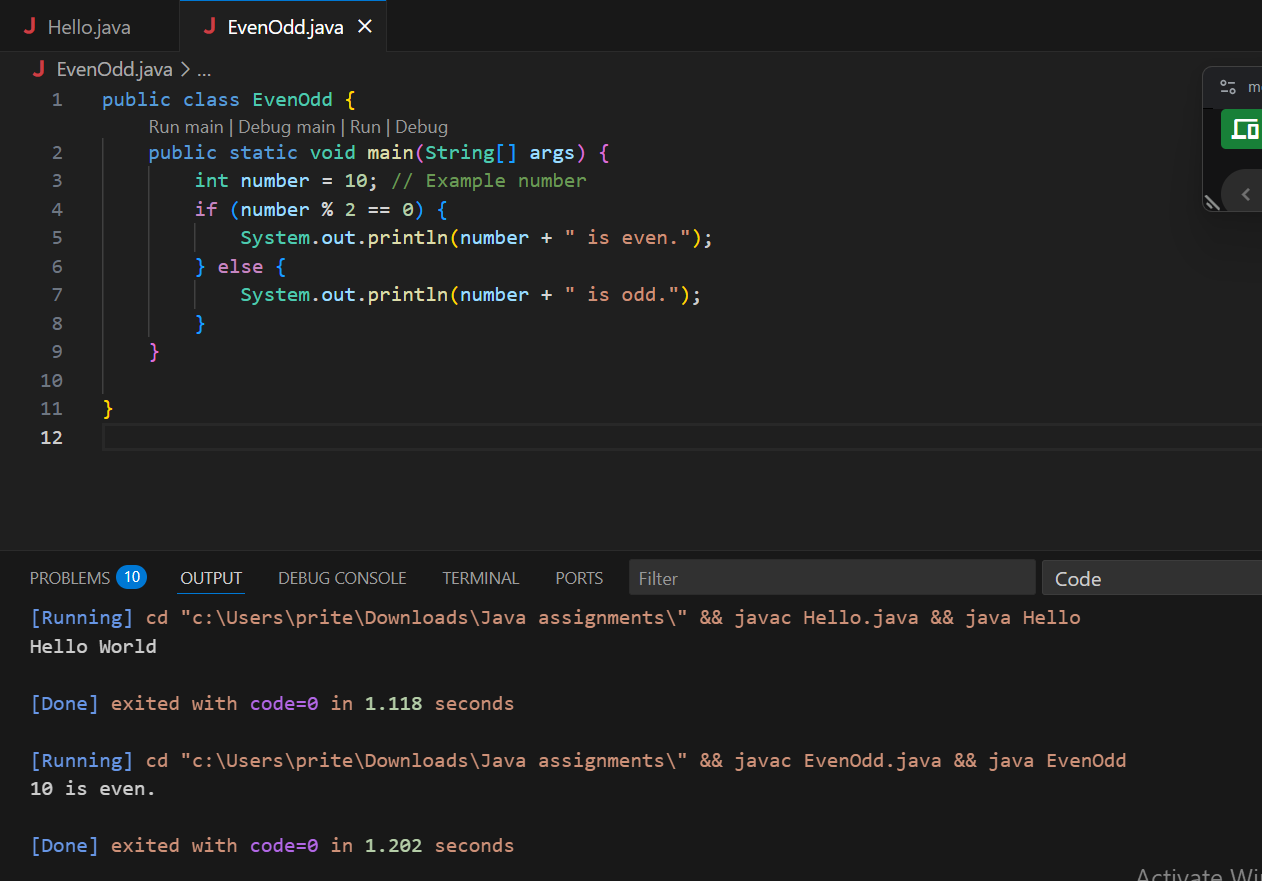
Executes one block of code if the condition is true, otherwise executes another block.

**3. switch Statement**

Used to select one option from multiple choices based on the value of a variable. Each option is called a "case".

## 9. Write a Java program to find whether a number is even or odd.

import java.util.Scanner;  
public class EvenOdd {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.in);  
 int num = sc.nextInt();  
 if (num % 2 == 0)  
 System.out.println("Even");  
 else  
 System.out.println("Odd");  
 }  
}



## 10. What is the difference between while and do-while loop?

while: checks condition before executing, May never execute  
 do-while: executes at least once, Executes at least once

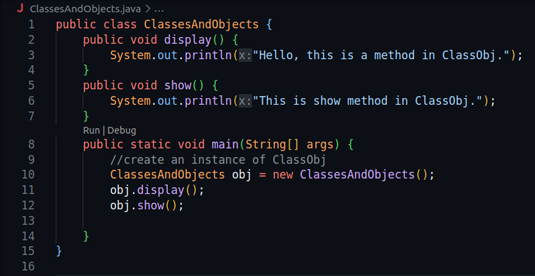
**Object-Oriented Programming (OOPs)**

## 1. What are the main principles of OOPs in Java?

* **Encapsulation**: Data hiding using classes
* **Abstraction**: Hiding implementation details
* **Inheritance**: Code reuse through subclasses
* **Polymorphism**: Many forms of methods/objects

## 2. What is a class and an object in Java? Give examples.

Class: Blueprint of object  
Object: Instance of class  
Example:  
class Car { String color; void drive() {} }



## 3. Write a program using class and object to calculate area of a rectangle.

class Rectangle {

int length, breadth;

int calculateArea() {

return length \* breadth;

}

}

public class Main {

public static void main(String[] args) {

Rectangle r = new Rectangle();

r.length = 10;

r.breadth = 5;

System.out.println("Area: " + r.calculateArea());

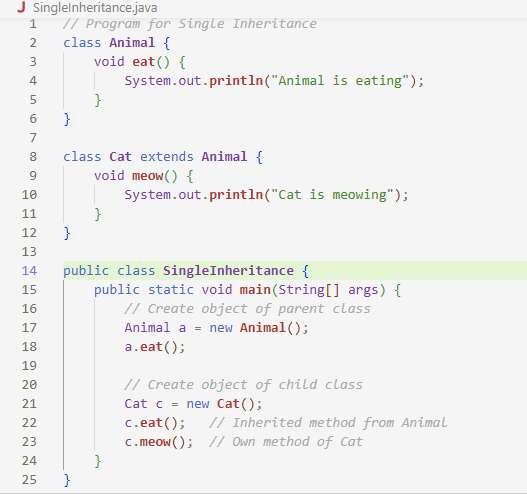
}

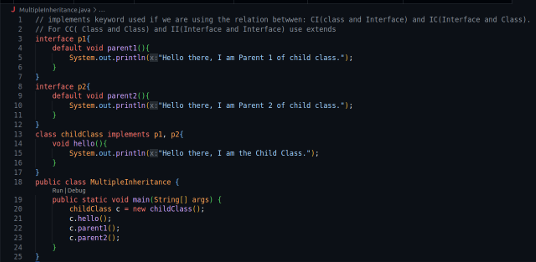
}

## 

## 4. Explain inheritance with real-life example and Java code.

Inheritance is an OOP principle where one class (child) inherits the properties and behaviors of another class (parent). It promotes code reusability and supports method overriding.





## 5. What is polymorphism? Explain with compile-time and runtime examples.

Polymorphism: Same method behaves differently.  
Compile-time: Method Overloading  
Runtime: Method Overriding

Polymorphism means "many forms". In Java, it allows one interface, method, or object to behave in different ways. It is a key concept of Object-Oriented Programming (OOP).

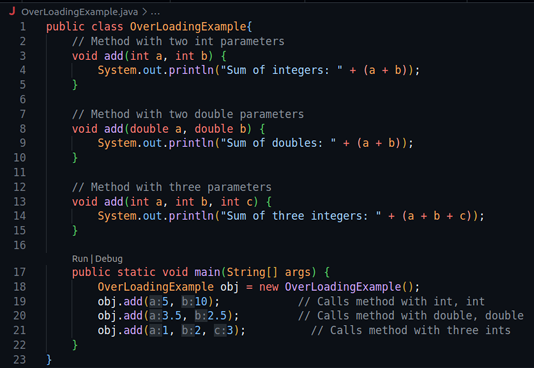
There are two types of polymorphism in Java:

* Compile-time Polymorphism (Method Overloading)

1. It occurs when multiple methods in the same class have the same name but different parameters.
2. The method to be called is decided at compile time

## 6. What is method overloading and method overriding? Show with examples.

Overloading: Same method, different params  
Overriding: Subclass modifies superclass method



## 7. What is encapsulation? Write a program demonstrating encapsulation.

Encapsulation is the process of binding data (variables) and code (methods) into a single unit, typically a class, and restricting direct access to some of the object’s components. It is achieved by:

* Making variables private
* Providing public getter and setter methods to access and modify those variables

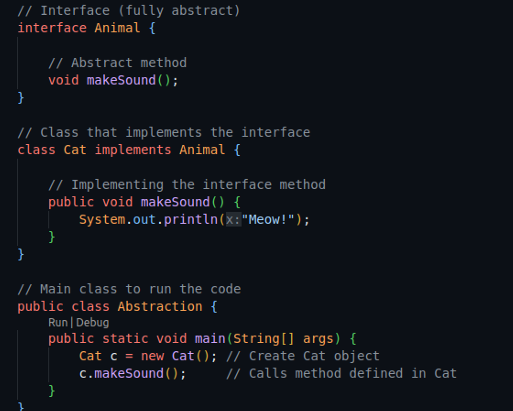
Benefits of Encapsulation:

* Protects data from unauthorized access
* Increases code maintainability and flexibility
* Makes the class easier to use and modify



## 8. What is abstraction in Java? How is it achieved?

Abstraction hides implementation details.  
Achieved via abstract classes and interfaces



## 9. Explain the difference between abstract class and interface.

Abstract Class in Java:

An abstract class is a class that is declared with the abstract keyword.

It can have:

* Abstract methods (without a body)
* Concrete methods (with a body)

It is used when you want to provide a base class with some shared code and some methods that must be implemented by child classes.

* Supports partial abstraction
* Can have constructors
* Can have instance variables
* Can be inherited using extends
* A class can extend only one abstract class

**Interface in Java:**

An interface is a blueprint of a class that contains only abstract methods (by default) and constants.

It is used to define a set of rules or behaviors that multiple unrelated classes can follow.

* Supports full abstraction (100%)
* All methods are abstract and public by default
* Variables are public, static, and final
* No constructors allowed
* A class can implement multiple interfaces

10. Create a Java program to demonstrate the use of interface.