JAVA ASSIGNMENT-1 (7PM BATCH)

Ques. 1. When was java develop and who is the inventor of java? Why we use java for programming?

Ans. 1.Java was developed in 1995 by Sun Microsystems. The inventor of Java was James Gosling.

Java is a high level programming language used for various applications due to its:

* Platform Independence**:** It runs on any system with a Java Virtual Machine (JVM).
* Object-Oriented Nature**:** Promotes code reusability and maintainability.
* Robustness**:** Prevents common errors with strong type checking and exception handling.
* Security**:** Designed with security features for network environments.
* Large Ecosystem**:** Offers a wide range of libraries and frameworks.
* Performance**:** Modern JVMs have improved execution speed.

Ques. 2. What are the features of java.

Ans. 2.

* Platform-independent.
* Object-oriented.
* Robust.
* Inheritance.
* Secure.
* Automatic.
* Multithreaded.

Ques. 3. what is JDK, JRE, JVM?

Ans. 3.

* JDK (Java Development Kit): A complete package for developing Java applications, including the JRE, compiler, debugger, and other tools.
* JRE (Java Runtime Environment): Essential for running Java programs, containing the JVM, class libraries, and other components.
* JVM (Java Virtual Machine**):** An abstract machine that executes Java bytecode, ensuring platform independence.

Ques. 4. what is the structure of java program. write a program to print hello world.

Ans. 4.

1. Package Declaration.
2. Import Statements:
3. Class Declaration:
4. Main Method**:**

Example: Printing "Hello World"

Java

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello World");

}

}

Ques. 5. Define Variables in java. Explain different scope of variable.

Ans. 5. Variables in Java are containers that store data values. They have a specific data type (e.g., int, double, String) that determines the kind of values they can hold.

1. Local Variables.

 Declared within a method, constructor, or block of code.

 Exist only within the scope where they are declared.

 Must be initialized before use.

 Cannot be accessed outside their scope.

1. Instance Variable.

 Declared within a class but outside any method.

 Belong to individual objects of the class.

 Can be accessed directly from within the class's methods.

 Have default values if not explicitly initialized.

1. Static Variables.

 Declared within a class using the static keyword.

 Belong to the class itself, not individual objects.

 Can be accessed using the class name and a dot.

 There's only one copy of a class variable shared by all objects of the class.

Ques. 6. Define data types in java.

Ans. 6. Data types in Java define the kind of values a variable can hold and the operations that can be performed on them. They are categorized into two main groups:

1. Primitive Data Types:

* Numeric: byte, short, int, long, float, double
* Character: char
* Boolean: boolean

2. Non Primitive Data Types:

* Classes**:** Blueprints for creating objects.
* Interfaces: Define contracts that classes must implement.
* Arrays: Ordered collections of elements of the same data type.

Ques. 7. What is type casting in java?

Ans. 7. Type casting in Java is the process of converting one data type to another. It's essential when you need to perform operations on different data types or store a value of one type in a variable of another type.

Ques. 8. Write a program to add two number by taking input from user.

Java

Import Java.util.Scanner;

Public class add\_twoNo.{

Public Static void main(String[ Args]){

Int a,b ,Sum;

Scanner sc=new Scanner(System.in);

System.out.println(“Enter the first no”);

a=sc.nextInt();

System.out.println(“Enter the Second no”);

b=sc.nextInt();

Sum=a+b;

System.out.println(Sum);

}

}

Ques. 9. Define operator and its type?

Ans. 9. An Operator is a special symbol or keyword that performs operations on one or more operands.

 **Arithmetic Operators**

* **Addition (+)**: 5 + 3 results in 8.
* **Subtraction (-)**: 7 - 2 results in 5.
* **Multiplication (\*)**: 4 \* 6 results in 24.
* **Division (/)**: 8 / 2 results in 4.
* **Modulus (%)**: 10 % 3 results in 1.

 **Relational Operators**

* **Equal to (==)**: 5 == 5 results in true.
* **Not equal to (!=)**: 5 != 3 results in true.
* **Greater than (>)**: 7 > 4 results in true.
* **Less than (<)**: 3 < 5 results in true.
* **Greater than or equal to (>=)**: 5 >= 5 results in true.
* **Less than or equal to (<=)**: 4 <= 6 results in true.

 **Logical Operators**

* **AND (&&)**: (true && false) results in false.
* **OR (||)**: (true || false) results in true.
* **NOT (!)**: !true results in false.

 **Assignment Operators**

* **Assignment (=)**: x = 10.
* **Addition assignment (+=)**: x += 5.
* **Subtraction assignment (-=)**: x -= 3.
* **Multiplication assignment (\*=)**: x \*= 4.
* **Division assignment (/=)**: x /= 2.
* **Modulus assignment (%=)**: x %= 3.

 **Bitwise Operators**

* **AND (&)**: 5 & 3 results in 1.
* **OR (|)**: 5 | 3 results in 7.
* **XOR (^)**: 5 ^ 3 results in 6.
* **NOT (~)**: ~5 results in -6.
* **Left shift (<<)**: 2 << 1 results in 4.
* **Right shift (>>)**: 4 >> 1 results in 2.

 **Conditional (Ternary) Operator**

* **Ternary (?:)**: (a > b) ? a : b.

 **Unary Operators**

* **Unary plus (+)**: +5 results in 5.
* **Unary minus (-)**: -5 results in -5.
* **Increment (++)**: x++.
* **Decrement (--)**: x--.
* **Logical NOT (!)**: !true results in false.

Ques. 10. Write a program individually to perform operators operation.