JAVA VARIABLES AND DATA TYPES ASSIGNMENT-5

Q.1 what is statically type and dynamic type programming language

Ans.

- Statically typed languages: Statically typed languages perform type checking at compile time. This means that the compiler verifies that all variables are assigned to values of the correct type before the program is executed. This can help to prevent a wide range of errors, such as trying to add a string to a number or calling a function with the wrong arguments.
- Dynamically typed languages: Dynamically typed languages perform type checking at runtime. This means that the program checks the type of variables and expressions as they are being executed. This can be more flexible than static typing, but it can also lead to errors that are not detected until the program is running.

Q.2 what is the variable in Java

Ans. A variable in Java is a named location in memory that can store a value. Variables are used to store data that is used by a Java program.

Q.3 how to assign value in value?

Ans. To assign a value in Java, you use the equals sign (=). For example, the following code assigns the value 10 to the variable x:

```
int x = 10;
```

Q.4 what is the primitive data type?

Ans.
A primitive data type in Java is a predefined data type that is used to store basic values. Primitive data types are built into the Java language and cannot be changed.

There are eight primitive data types in Java:

- byte: A byte is an 8-bit signed integer, with a range of -128 to 127.
- short: A short is a 16-bit signed integer, with a range of -32,768 to 32,767.
- int: An int is a 32-bit signed integer, with a range of -2,147,483,648 to 2,147,483,647.
- long: A long is a 64-bit signed integer, with a range of -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.
- float: A float is a 32-bit floating-point number, with a range of approximately -3.402823466E+38 to 3.402823466E+38.
- double: A double is a 64-bit floating-point number, with a range of approximately -1.7976931348623157E+308 to 1.7976931348623157E+308.
- char: A char is a 16-bit Unicode character.
- boolean: A boolean is a single bit that can store either true or false.

Q.5 what are the identifier in Java?

Ans.

An identifier in Java is a name that is given to a class, variable, method, package, or interface. Identifiers are used to identify elements of a Java program and to distinguish them from each other.

name

age

myVariable

calculateArea()

com.example.myapp

Q.6 list of operator in Java?

Ans.
There are eight types of operators in Java:

- Arithmetic operators: Arithmetic operators are used to perform arithmetic operations on primitive data types. The following are the arithmetic operators in Java:
    - +: Addition
    - -: Subtraction
    - *: Multiplication
    - /: Division
    - %: Modulo (remainder)
- Assignment operators: Assignment operators are used to assign values to variables. The following are the assignment operators in Java:
    - =: Simple assignment
    - +=: Addition assignment
    - -=: Subtraction assignment
    - *=: Multiplication assignment
    - /=: Division assignment
    - %=: Modulo assignment
- Logical operators: Logical operators are used to perform logical operations on Boolean expressions. The following are the logical operators in Java:
    - &&: AND
    - ||: OR
    - !: NOT

- Relational operators: Relational operators are used to compare two values and return a Boolean value indicating whether the comparison is true or false. The following are the relational operators in Java:
    - ==: Equal to
    - !=: Not equal to
    - <: Less than
    - >: Greater than
    - <=: Less than or equal to
    - >=: Greater than or equal to
- Bitwise operators: Bitwise operators are used to perform bit-level operations on primitive data types. The following are the bitwise operators in Java:
    - &: AND
    - |: OR
    - ^: XOR
    - ~: NOT
    - <<: Left shift
    - >>: Right shift
    - >>>: Unsigned right shift
- Conditional operators: Conditional operators are used to evaluate a Boolean expression and return a different value depending on the result of the evaluation. The following is the conditional operator in Java:
    - ?:: Ternary operator
- Instanceof operator: The instanceof operator is used to check whether an object is an instance of a particular class.
- Type comparison operator: The instanceof operator can also be used to compare two types.

Q.7 explain about increment and decrement operator with example ?

Ans.

The increment (++) and decrement (--) operators in Java are unary operators, meaning they operate on a single operand. They are used to increase or decrease the value of an integer, floating-point, or character variable by 1. These operators can be applied in two ways: prefix and postfix.

Prefix increment/decrement: When used as a prefix, the ++ or -- operator is applied to the operand before the expression is evaluated. For example, the following code increments the value of the variable x by 1 before it is printed to the console:

Java

```
int x = 5;

System.out.println(++x); // Prints 6
```

Postfix increment/decrement: When used as a postfix, the ++ or -- operator is applied to the operand after the expression is evaluated. For example, the following code prints the value of the variable x before it is incremented by 1:

Java

```
int x = 5;

System.out.println(x++); // Prints 5
```

The difference between prefix and postfix increment/decrement operators is subtle but important. Prefix increment/decrement operators are often used to modify the value of a variable before it is used in an expression. Postfix increment/decrement operators are often used to modify the value of a variable after it has been used in an expression.

Here are some more examples of increment and decrement operators:

Java

```
int y = 10;


// Prefix increment

System.out.println(++y); // Prints 11


// Postfix increment

System.out.println(y++); // Prints 11


// Prefix decrement

System.out.println(--y); // Prints 10


// Postfix decrement

System.out.println(y--); // Prints 10
```