**1) What is statically typed Dynamically typed Programming Language ?**

In programming languages, the terms "statically typed" and "dynamically typed" describe the way in which a language manages the type of its variables.

A statically typed language is one in which the type of a variable must be declared when it is created and cannot change afterwards. The type information is checked at compile-time and, if there is a type mismatch, a compile-error is generated. Examples of statically typed languages include Java, C++, and C#.

A dynamically typed language is one in which the type of a variable is determined at runtime. This means that a variable can change its type over its lifetime and type checking is done at runtime, not at compile-time. Examples of dynamically typed languages include Python, Ruby, and JavaScript.

Dynamic typing can allow for more flexible and faster development, as the programmer does not need to specify the type of a variable upfront and can change it at runtime if needed. However, it can also lead to unexpected behavior and errors if the type of a variable is not properly managed. Statically typed languages, on the other hand, can provide more safety and predictability, as the type information is checked before the program runs.

**2) What is Variable in java ?**

In Java, a variable is a named storage location in the memory of a computer, which is used to store data values. A variable must be declared with its type (such as int, double, or String) and a unique name, and then it can be used to store values throughout the code.

**Here's an example of how to declare and use variables in Java:**

**code**

int age; // declaration of a variable "age" of type int

age = 30; // assignment of the value 30 to the variable "age"

System.out.println(age); // prints the value of "age", which is 30

Variables in Java can be of different data types, such as numeric, character, and boolean. The type of a variable determines the kind of values it can store and the operations that can be performed on those values.

**3) How to Assign the value to the variable in java ?**

In Java, you can assign a value to a variable using the assignment operator (=). Here's an example:

**code**

int age; // declaration of a variable "age" of type int

age = 30; // assignment of the value 30 to the variable "age"

In this example, the statement age = 30 assigns the value 30 to the variable age. After this assignment, the value of age is 30. You can then use the value of age in your code by referring to the variable age.

It's important to note that you must declare a variable before you can assign a value to it. The declaration specifies the type and name of the variable. In this example, int age declares a variable of type int with the name age.

**4) What are the primitive data types in java ?**

In Java, there are eight primitive data types:

**1) byte:** 8-bit signed two's complement integer

**2) short:** 16-bit signed two's complement integer

**3) int:** 32-bit signed two's complement integer

**4) long:** 64-bit signed two's complement integer

**5) float:** 32-bit single-precision floating-point number

**6) double:** 64-bit double-precision floating-point number

**7) char:** 16-bit Unicode character

**8)boolean:** a binary value that can be either true or false

These primitive data types are the basic building blocks of Java and are used to store values in the memory of a computer. The data type of a variable determines the size of memory it requires and the type of operations that can be performed on it.

**5) What are Identifiers in java?**

In Java, an identifier is a name given to a variable, method, class, or any other user-defined item. Identifiers are used to represent the elements of a program and must follow certain rules in order to be considered valid:

Must start with a letter, dollar sign ($), or underscore (\_).

Subsequent characters can be letters, digits, dollar signs, or underscores.

Cannot be a Java reserved word (such as int, class, if, etc.).

Cannot contain whitespace characters.

Are case-sensitive (age and Age are considered different identifiers).

**Here are some examples of valid Java identifiers:**

**code**

firstName

\_private

$salary

student\_id

It's a good practice to choose descriptive and meaningful names for your identifiers to make your code more readable and understandable.

**6) List the Operators in java ?**

In Java, there are several types of operators that can be used to perform operations such as arithmetic, relational, logical, bitwise, etc. Some of the commonly used operators are:

**Arithmetic operators**: +, -, \*, /, % (modulus)

**Relational operators**: <, >, <=, >=, ==, !=

**Logical operators**: &&, ||, !

**Bitwise operators:** &, |, ^, ~, &#8203;oaicite:{"index":0,"invalid\_reason":"Malformed citation <<, >>, >>>"}&#8203;

**Assignment operators**: =, +=, -=, \*=, /=, %=, &=, |=, ^=, &#8203;oaicite:{"index":1,"invalid\_reason":"Malformed citation <<=, >>=, >>>"}&#8203;=

**Ternary operator:** ? :

**Increment and decrement operators:** ++, --

**Instanceof operator:** instanceof

**Conditional operator:** (condition) ? expression1 : expression2

It's important to understand the behavior and precedence of each operator in order to write correct and efficient code.

**7) Explain about the increment and Decrement Operator and give one examples?**

The increment operator (++) increases the value of its operand by 1, while the decrement operator (--) decreases the value of its operand by 1.

The increment and decrement operators can be used in both prefix and postfix form. In the prefix form, the operator is

placed before the operand, while in the postfix form, the operator is placed after the operand. The difference between prefix and postfix form is that in the prefix form, the value is incremented or decremented before its value is used, while in the postfix form, the value is incremented or decremented after its value is used.

**Example:**

**code**

int a = 10;

int b = 20;

**// Prefix form**

++a; // a = 11

--b; // b = 19

**// Postfix form**

a++; // a = 12

b--; // b = 18