1. What is Statically Typed and Dynamically Typed Programming Language?

Statically Typed: In statically typed languages, the data type of a variable is known at compile-time, and it must be explicitly declared. Examples include Java, C++, and Go.
Example:
java

```
int x = 5; // Type (int) must be declared before using the variable.
```

-

Dynamically Typed: In dynamically typed languages, the data type of a variable is determined at runtime, and explicit declaration is not required. Examples include Python and JavaScript.
Example:
python

```
x = 5  # No type declaration required; type is determined at runtime.
```

-

---

2. What is a Variable in Java?

A variable in Java is a named memory location used to store a value. It acts as a container for data that can be accessed and manipulated during program execution.
Example:

java
Copy code

```
int age = 25; // `age` is the variable storing the value 25.
```

---

3. How to Assign a Value to a Variable?

In Java, you can assign a value to a variable using the assignment operator `=`.
Example:

java

```
int num;          // Declare a variable
num = 10;         // Assign a value
String name = "John"; // Declare and assign a value in one line
```

4. What are Primitive Data Types in Java?

Java provides 8 primitive data types for storing simple values:

1. **byte**: 1 byte (8 bits), range: -128 to 127
2. **short**: 2 bytes (16 bits), range: -32,768 to 32,767
3. **int**: 4 bytes (32 bits), range: -2,147,483,648 to 2,147,483,647
4. **long**: 8 bytes (64 bits), range: -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
5. **float**: 4 bytes, stores fractional numbers with up to 6-7 decimal digits
6. **double**: 8 bytes, stores fractional numbers with up to 15 decimal digits
7. **char**: 2 bytes, stores a single character
8. **boolean**: 1 bit, stores true or false

5. What are the Identifiers in Java?

Identifiers are the names used to identify variables, methods, classes, or other program elements.
Rules for Identifiers:

- Must begin with a letter, underscore (_), or dollar sign ($).
- Cannot be a reserved keyword.
- Must not contain spaces or special characters except `_` and `$`.
- Can be of any length.
  Example:

java

```
int age;       // `age` is an identifier for a variable
String name;   // `name` is an identifier for a variable
```

6. List the Operators in Java

Java supports the following operators:

- **Arithmetic Operators**: `+`, `-`, `*`, `/`, `%`
- **Relational/Comparison Operators**: `==`, `!=`, `<`, `>`, `<=`, `>=`
- **Logical Operators**: `&&`, `||`, `!`

- **Bitwise Operators**: `&`, `|`, `^`, `~`, `<<`, `>>`, `>>>`
- **Assignment Operators**: `=`, `+=`, `-=`, `*=`, `/=`, `%=`
- **Unary Operators**: `+`, `-`, `++`, `--`
- **Ternary Operator**: `condition ? expression1 : expression2`
- **Instanceof Operator**: `object instanceof class`

---

7. Explain Increment and Decrement Operators with Examples

- **Increment Operator (`++`)**: Increases the value of a variable by 1.
    - **Pre-Increment (`++x`)**: Value is incremented before use.
    - **Post-Increment (`x++`)**: Value is incremented after use.
      Example:

java

```java
int x = 5;
System.out.println(++x); // Output: 6 (Pre-increment)
System.out.println(x++); // Output: 6 (Post-increment)
System.out.println(x);   // Output: 7
```

-
- **Decrement Operator (`--`)**: Decreases the value of a variable by 1.
    - **Pre-Decrement (`--x`)**: Value is decremented before use.
    - **Post-Decrement (`x--`)**: Value is decremented after use.
      Example:

java

```java
int y = 5;
System.out.println(--y); // Output: 4 (Pre-decrement)
System.out.println(y--); // Output: 4 (Post-decrement)
System.out.println(y);   // Output: 3
```

-

4o