DATA TYPES AND VARIABLES IN JAVA

1. Primitive Data Types :

Primitive data types are the most basic types of data that are predefined in Java.
These types represent simple values such as numbers and characters.

List of Primitive Data Types :

1. **byte** :
   - size : 1 byte (8 bits)
   - Range : -128 to 127
   - Default value : 0
   - Example : byte a = 100;

2. **short** :
   - size : 4 bytes (32 bits)
   - Range : -2,147,483,648 to 2,147,483,647
   - Default value : 0
   - Example : short b = 32000;

3. **int** :
   - size : 4 bytes (32 bits)
   - Range : -2,147,483,648 to 2,147,483,647
   - Default value : 0
   - Example : int c = 1000;

4. long:
- size : 8 bytes (64 bits)
- Range : -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.
- Default value : 0L
- Example : long d = 1234567890L;

5. float:
- size : 4 bytes (32 bits)
- Range : used for decimal values (single precision)
- Default value : 0.0f
- Example : float e = 3.14f;

6. double:
- size : 8 bytes (64 bits)
- Used for : Decimal values (double precision)
- Default value : 0.0d
- Example : double f = 3.14159;

7. char:
- size : 2 bytes (16 bits)
- Used for : single characters
- Default value : '\u0000' (null character)
- Example : char g = 'A';

8. boolean:
- size : 1 bit
- Used for : True/False values
- Default value : false
- Example : boolean h = true;

@curious_programmer

2. Non-Primitive Data Types:

Non-Primitive data types, also known as reference types, are used to store more complex structures. These are objects created from classes.

2.1. Strings:

- **String** is a sequence of characters, and it is represented as an object in Java.
- Strings are immutable, meaning that their content cannot be changed once created.

Example:

```
String message = "Hello, Java!";
```

String operations can include:

- Concatenation : "Hello" + " " + "World"
- Length : message.length()
- Substring : message.substring(0, 5)

2.2. Arrays :

- An **Array** is a collection of variables of the same type, stored in contiguous memory locations.

- Arrays have fixed size after they are created.

Example :

```
int [] numbers = {1, 2, 3, 4, 5};
string [] names = {"Alice", "Bob", "Charlie"};
```

You can access elements using an index (starting from 0) :

```
system.out.println(numbers[0]); // output : 1
```

3. Variables and Constants :

3.1. Variables :

- A **variable** is a container for storing data. It has a name, a type, and a value.
- Variables are declared by specifying the type followed by the name.

Syntax:

```
<datatype> <variable_name> = <value>;
```

Example:

```
int age = 25;
String name = "John";
```

3.2. Constants :

- A **constant** is a variable whose value cannot be changed once it is assigned.
- Constants are declared using the final keyboard.

Syntax :

```
final <datatype> <constant-name> = <value> ;
```

Example :

```
final int MAX-VALUE = 100 ;
```

@ Curious_.programmer

4. Type Casting and Type Conversion :

4.1. Type Casting:

Type casting refers to converting one data type to another. In Java, there are two types of casting :

- **Widening (Automatic) :**

Converting a smaller data type to a larger one. It happens automatically.

  - Example : int to long, float to double

- **Narrowing (Manual) :**

Converting a larger data type to a smaller one. It requires explicit casting.

  - Example : double to int

@ Curious-programmer

Examples :

```
int i = 10;
long l = i; // widening Casting (int to long)
```

```
double d = 3.14;
int j = (int) d; // Narrowing Casting (double to int)
```

4.2. Type Conversion:

Type conversion refers to changing the type of variable from one data type to another.

This can be done in two ways:

- Implicit Conversion (Widening)
- Explicit Conversion (Narrowing)

Example:

```
String str = "123";
int num = Integer.parseInt(str);
                    // string to int
```

@Curious_programmer

5. Unicode System :

- **Unicode** is a character encoding standard that is used to represent text and symbols in Java
- Each character is assigned a unique Unicode code point.
- The **char** data type in Java stores a single Unicode character.

Example :

```
char letter = 'A' ; // Unicode for 'A' is 65

char symbol = '\u0041' ; //Unicode representation
                           Of 'A'
```

@ Curious-.programmer