Data Types - Data types in Java are used to specify the type of data that a variable can hold. They determine the size and range of values that can be stored in a variable.

There are generally two data types –

1. Primitive or Predefined
2. Non-Primitive or User defined.

Primitive Data Types - Primitive data types are the basic data types that are predefined in the Java programming language. They represent a single value and are directly supported by the Java Virtual Machine (JVM). The primitive data types in Java can be divided into the following subparts:

Numeric Data Types

- **Integral Types:**

1. **byte:** The byte data type is an 8-bit signed two's complement integer, which means it can store whole numbers from -128 to 127.

   Example: byte age = 25;
   Size: 1 byte (8 bits)
   Range: -2^7 (-128) to 2^7 - 1 (127)

2. **short:** The short data type is a 16-bit signed two's complement integer, which means it can store whole numbers from -32,768 to 32,767.

   Example: short yearOfBirth = 1990;
   Size: 2 bytes (16 bits)
   Range: -2^15 (-32,768) to 2^15 - 1 (32,767)

3. **int:** The int data type is a 32-bit signed two's complement integer, which means it can store whole numbers from -2,147,483,648 to 2,147,483,647.

   Example: int population = 7900000;
   Size: 4 bytes (32 bits)
   Range: -2^31 (-2,147,483,648) to 2^31 - 1 (2,147,483,647)

4. **long:** The long data type is a 64-bit signed two's complement integer, which means it can store whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.

   Example: long earthPopulation = 7_900_000_000L;
   Size: 8 bytes (64 bits)

Range: -2^63 (-9,223,372,036,854,775,808) to 2^63 - 1 (9,223,372,036,854,775,807)

- **Character Type:**

1. **char:** The char data type is a single 16-bit Unicode character, which means it can store a wide range of characters, including Latin, Greek, Cyrillic, and many other scripts.

   Example: char letter = 'A';
   Size: 2 bytes (16 bits)
   Range: '\u0000' (0) to '\uffff' (65,535)

- **Floating-Point Types:**

1. **float:** The float data type is a 32-bit IEEE 754 floating-point number, which means it can store decimal values.

   Example: float pi = 3.14159F;
   Size: 4 bytes (32 bits)
   Range: Approximately ±3.40282347E+38 (6-7 significant digits)

2. **double:** The double data type is a 64-bit IEEE 754 floating-point number, which means it can store decimal values with higher precision than the float data type.

   Example: double gravity = 9.80665;
   Size: 8 bytes (64 bits)
   Range: Approximately ±1.79769313486231570E+308 (15-16 significant digits)

- **Boolean Data Type**

boolean: The boolean data type represents a single bit of information, storing only two possible values: true and false.

Example: boolean isStudent = true;
Size: The size of a boolean value is implementation-dependent, but it is typically 1 bit.
Range: The boolean data type can only represent two values: true or false.

Non-Primitive Data Types - Non-primitive data types, also known as reference data types, are more complex data types that are created by the programmer. They are used to store collections of data or more complex objects. The non-primitive data types in Java can be divided into the following subparts:

Reference Types:

String: The String data type represents a sequence of characters. Strings are immutable, meaning their values cannot be changed once they are created.

```
String name = “Kunal Yadav”
String message = “Hello World”
```

Array: It is a collection of elements of same data type. An array can be dynamically created as well as of fixed size.

```
int[] numbers = {1, 2, 3, 4, 5};
String[] names = {”John”, ”Jane”, ”Bob”, ”Alice”};
```

Classes: Classes are user-defined data types that can have fields (variables) and methods. Classes are the building blocks of object-oriented programming in Java.

Interfaces: Interfaces define a contract that a class must implement. They provide a way to achieve abstraction and define a common set of methods that multiple classes can implement.

Enumeration Type:

Enum: Enum is a special data type in Java that allows you to define a set of named constants. Enums are useful for representing a fixed set of options or choices

```
. public enum DayOfWeek {

    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY,
    SUNDAY
}

DayOfWeek today = DayOfWeek.MONDAY;
```