Basic Datatypes

Variables are nothing but ***reserved memory locations to store values***. This means that when you create a variable you reserve some space in the memory.

Based on the data type of a variable, the operating system allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different datatypes to variables, you can store integers, decimals, or characters in these variables.

There are two data types available in Java:

- Primitive Datatypes
- Reference/Object Datatypes

Primitive Datatypes

There are **eight** primitive datatypes **supported by Java**. Primitive datatypes are **predefined** by the language and named by a keyword. Let us now look into the eight primitive data types in detail.

byte:

- Byte data type is an 8-bit signed two's complement integer
- Minimum value is -128 (-2^7)
- Maximum value is 127 (inclusive)(2^7 -1)
- Default value is 0
- Byte datatype is used to save space in large arrays, mainly in place of integers, since a byte is four times smaller than an integer
- Example: byte a = 100 , byte b = -50

short:

- Short datatype is a 16-bit signed two's complement integer
- Minimum value is -32,768 (-2^15)
- Maximum value is 32,767 (inclusive) (2^15 -1)
- Short datatype can also be used to save memory as byte data type. A short is 2 times smaller than an integer
- Default value is 0
- Example: short s = 10000, short r = -20000

int:

- Int datatype is a 32-bit signed two's complement integer
- Minimum value is - 2,147,483,648 (-2^31)
- Maximum value is 2,147,483,647(inclusive) (2^31 -1)
- Integer is generally used as the default data type for integral values unless there is a concern about memory.
- The default value is 0
- Example: int a = 100000, int b = -200000

long:

- Long datatype is a 64-bit signed two's complement integer
- Minimum value is -9,223,372,036,854,775,808 (-2^63)
- Maximum value is 9,223,372,036,854,775,807 (inclusive) (2^63 -1)
- This type is used when a wider range than int is needed
- Default value is 0L
- Example: long a = 100000L, long b = -200000L

float:

- Float datatype is a single-precision 32-bit IEEE 754 floating point
- Float is mainly used to save memory in large arrays of floating point numbers
- Default value is 0.0f
- Float datatype is never used for precise values such as currency
- Example: float f1 = 234.5f

double:

- double datatype is a double-precision 64-bit IEEE 754 floating point
- This datatype is generally used as the default data type for decimal values, generally the default choice
- Double datatype should never be used for precise values such as currency
- Default value is 0.0d

boolean:

- boolean datatype represents one bit of information
- There are only two possible values: true and false

- This datatype is used for simple flags that track true/false conditions
- Default value is false
- Example: boolean one = true

char:

- char datatype is a single 16-bit Unicode character
- Minimum value is '\u0000' (or 0)
- Maximum value is '\uffff' (or 65,535 inclusive)
- Char datatype is used to store any character
- Example: char letterA ='A'

Reference Datatypes

- Reference variables are created using **defined constructors of the classes**. They are used to access objects. These variables are declared to be of a specific type that cannot be changed.
  For example, Employee, Puppy, etc.
- Class objects and various type of array variables come under reference datatype.
- **Default** value of any reference variable is **null**.
- A reference variable can be used to refer any object of the declared type or any compatible type.
  Example: Animal animal = new Animal("giraffe");

Variable Types

A variable provides us with named storage that our programs can manipulate.
Each variable in Java has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

You **must** declare all variables before they can be used.

Example:

```
int a, b, c; // Declares three ints, a, b, and c.
int a = 10, b = 10; // Example of initialization
byte B = 22; // initializes a byte type variable B.
double pi = 3.14159; // declares and assigns a value of PI.
char a = 'a'; // the char variable a iis initialized with value 'a'
```

Examples :

Local Variables

```
public class Test {
	public void pupAge() {
		int age = 0;
		age = age + 7;
		System.out.println("Puppy age is : " + age);
	}

	public static void main(String args[]) {
		Test test = new Test();
		test.pupAge();
	}
}
```

This will produce the following result:

```
Puppy age is: 7
```

Instance Variables

```
public class Product {
	public int Barcode;

	public static void main(String[] args) {

		Product prod1 = new Product();
		prod1.Barcode = 123456;

		Product prod2 = new Product();
		prod2.Barcode = 987654;

		System.out.println(prod1.Barcode);
		System.out.println(prod2.Barcode);
	}
}
```

This will produce the following result:

```
123456
987654
```

Class/static Variables

```
public class Product {
	public static int Barcode;

	public static void main(String[] args) {

		Product prod1 = new Product();
		prod1.Barcode = 123456;

		Product prod2 = new Product();
		prod2.Barcode = 987654;

		System.out.println(prod1.Barcode);
		System.out.println(prod2.Barcode);
	}
}
```

This will produce the following result:
987654
987654