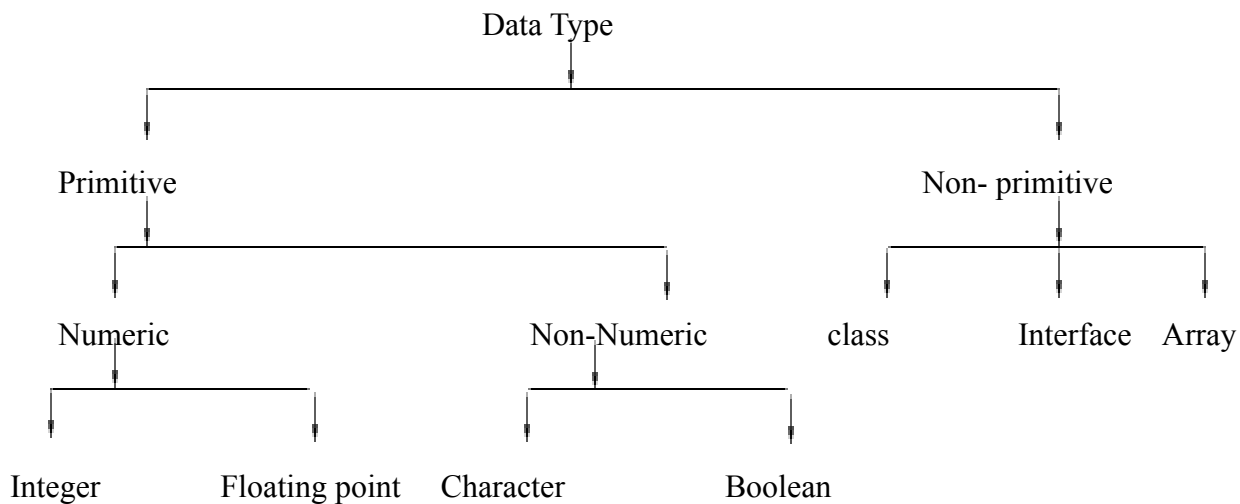


Data Types in Java

Every variable in java has a data type. Data types specify the size and type of values that can be stored in an identifier. Java language is rich in its data types. The variety of data types available allow the programmer to select the type appropriate to the need of application.

In java, data types are classified into two categories:

1. Primitive Data type or Intrinsic or built in data type
2. Non-Primitive Data type or derived or reference data type



Integer

This group includes byte, short, int, long

byte : It is 1 byte(8-bits) integer data type. Value range from -128 to 127. Default value zero.
example: byte b=10;

short : It is 2 bytes(16-bits) integer data type. Value range from -32768 to 32767. Default value zero. example: short s=11;

int : It is 4 bytes(32-bits) integer data type. Value range from -2147483648 to 2147483647. Default value zero. example: int i=10;

long : It is 8 bytes(64-bits) integer data type. Value range from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Default value zero. example: long l=100012;

Example:

```
public class Demo{

    public static void main(String[] args) {
        // byte type
        byte b = 20;
        System.out.println("b= "+b);

        // short type
        short s = 20;
        System.out.println("s= "+s);

        // int type
        int i = 20;
        System.out.println("i= "+i);

        // long type
        long l = 20;
        System.out.println("l= "+l);

    }
}
```

Floating-Point Number

This group includes float, double

float : It is 4 bytes(32-bits) float data type. Default value 0.0f. example: float ff=10.3f;

double : It is 8 bytes(64-bits) float data type. Default value 0.0d. example: double db=11.123;

```
public class Demo{

    public static void main(String[] args) {
        // float type
        float f = 20.25f;
        System.out.println("f= "+f);

        // double type
        double d = 20.25;
        System.out.println("d= "+d);
    }
}
```

```
} }
```

Characters

This group represent char, which represent symbols in a character set, like letters and numbers.

char : It is 2 bytes(16-bits) unsigned unicode character. Range 0 to 65,535.

example: char c='a';

```
public class Demo {  
  
    public static void main(String[] args) {  
  
        char ch = 'S';  
        System.out.println(ch);  
  
        char ch2 = '&';  
        System.out.println(ch2);  
  
        char ch3 = '$';  
        System.out.println(ch3);  
  
    }  
}
```

Boolean

Boolean type is used when we want to test a particular condition during the execution of the program. There are only two values that a Boolean type can take: true or false.

Remember, both these words have been declared as keyword. Boolean type is denoted by the keyword Boolean and uses only 1 bit of storage.

```
public class Demo {  
  
    public static void main(String[] args) {  
  
        boolean t = true;  
        System.out.println(t);  
  
        boolean f = false;
```

```
System.out.println(f);
```

```
}}
```

Type Casting in Java

Casting is a process of changing one type value to another type. In Java, we can cast one type of value to another type. It is known as type casting.

In Java, type casting is classified into two types,

1. Automatic type conversion or Implicit
2. Explicitly

1. Automatic type conversion or Implicit

Automatic Type casting take place when,

- the two types are compatible
- the target type is larger than the source type

byte → short → int → long → float → double

```
class Test
{
    public static void main(String[] args)
    {
        int i = 100;
        long l = i; //no explicit type casting required
        float f = l; //no explicit type casting required
        System.out.println("Int value "+i);
        System.out.println("Long value "+l);
        System.out.println("Float value "+f);
    }
}
```

```
}
```

2. Explicitly

When you are assigning a larger type value to a variable of smaller type, then you need to perform explicit type casting. If we don't perform casting then compiler reports compile time error.

double → float → long → int → short → byte

```
class Test
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        double d = 100.04;
```

```
        long l = (long)d; //explicit type casting required
```

```
        int i = (int)l; //explicit type casting required
```

```
        System.out.println("Double value "+d);
```

```
        System.out.println("Long value "+l);
```

```
        System.out.println("Int value "+i);
```

```
    }
```

}