JAVA PRIMITIVE DATA TYPE

Description

Not everything in Java is an object. There is a special group of data types (also known as primitive types) that will be used quite often in programming. For performance reasons, the designers of the Java language decided to include these primitive types. Java determines the size of each primitive type. These sizes do not change from one operating system to another. This is one of the key features of the language that makes Java so portable. Java defines eight primitive types of data: byte, short, int, long, char, float, double, and boolean. The primitive types are also commonly referred to as simple types which can be put in four groups

- **Integers:** This group includes byte, short, int, and long, which are for whole-valued signed numbers.
- **Floating-point numbers:** This group includes float and double, which represent numbers with fractional precision.
- **Characters:** This group includes char, which represents symbols in a character set, like letters and numbers.
- **Boolean:** This group includes boolean, which is a special type for representing true/false values.

Let’s discuss each in details:

byte

The smallest integer type is byte. It has a minimum value of -128 and a maximum value of 127 (inclusive). The byte data type can be useful for saving memory in large arrays, where the memory savings actually matters. Byte variables are declared by use of the byte keyword. For example, the following declares and initialize byte variables called b:

```
byte b =100;
```

short:

The short data type is a 16-bit signed two's complement integer. It has a minimum value of -32,768 and a maximum value of 32,767 (inclusive). As with byte, the same guidelines apply: you can use a short to save memory in large arrays, in situations where the memory savings actually matters. Following example declares and initialize short variable called s:

```
short s =123;
```

---

int:

The most commonly used integer type is int. It is a signed 32-bit type that has a range from –2,147,483,648 to 2,147,483,647. In addition to other uses, variables of type int are commonly employed to control loops and to index arrays. This data type will most likely be large enough for the numbers your program will use, but if you need a wider range of values, use long instead.

```
int  v = 123543;

int  calc = -9876345;
```

long:

long is a signed 64-bit type and is useful for those occasions where an int type is not large enough to hold the desired value. It has a minimum value of -9,223,372,036,854,775,808 and a maximum value of 9,223,372,036,854,775,807 (inclusive). Use of this data type might be in banking application when large amount is to be calculated and stored.

```
long amountVal = 1234567891;
```

float:

Floating-point numbers, also known as real numbers, are used when evaluating expressions that require fractional precision. For example interest rate calculation or calculating square root. The float data type is a single-precision 32-bit IEEE 754 floating point. As with the recommendations for byte and short, use a float (instead of double) if you need to save memory in large arrays of floating point numbers. The type float specifies a single-precision value that uses 32 bits of storage. Single precision is faster on some processors and takes half as much space as double precision. The declaration and initialization syntax for float variables given below, please note “f” after value initialization.

```
float intrestRate = 12.25f;
```

double:

Double precision, as denoted by the double keyword, uses 64 bits to store a value. Double precision is actually faster than single precision on some modern processors that have been optimized for high-speed mathematical calculations. All transcendental math functions, such as sin( ), cos( ), and sqrt( ), return double values. The declaration and initialization syntax for double variables given below, please note “d” after value initialization.

```
double sineVal = 12345.234d;
```

boolean:

The boolean data type has only two possible values: true and false. Use this data type for simple flags that track true/false conditions. This is the type returned by all relational operators, as in the case of a < b. boolean is also the type required by the conditional expressions that govern the control statements such as if or while.

```
boolean  flag = true;

booleanval  = false;
```

char:

In Java, the data type used to store characters is char. The char data type is a single 16-bit Unicode character. It has a minimum value of '\u0000' (or 0) and a maximum value of '\uffff' (or 65,535 inclusive). There are no negative chars.

```
char ch1 = 88; // code for X

char  ch2 = 'Y';
```

Primitive Variables can be of two types

(1) Class level (instance) variable:

It's not mandatory to initialize Class level (instance) variable. If we do not initialize instance variable compiler will assign default value to it. Generally speaking, this default will be zero or null, depending on the data type. Relying on such default values, however, is generally considered bad coding practice.The following chart summarizes the default values for the above data types.

Primitive Data Type	**Default Value 3.6**
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	false

(2) Method local variable:

Method local variables have to be initialized before using it. The compiler never assigns a default value to an un-initialized local variable. If you cannot initialize your local variable where it is declared, make sure to assign it a value before you attempt to use it. Accessing an un-initialized local variable will result in a compile-time error.

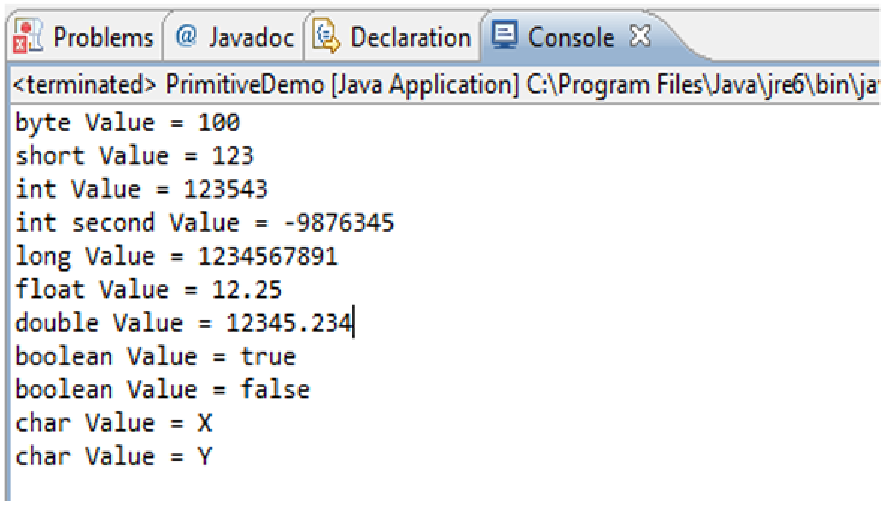
Let’s See simple java program which declares, initialize and print all above primitive types.

Java class PrimitiveDemo as below:

```java

package primitive;
public class PrimitiveDemo {
    public static void main(String[] args) {
        byte b =100;
        short s =123;
        int v = 123543;
        int calc = -9876345;
        long amountVal = 1234567891;
        float intrestRate = 12.25f;
        double sineVal = 12345.234d;
        boolean flag = true;
        boolean val = false;
        char ch1 = 88; // code for X
        char ch2 = 'Y';
        System.out.println("byte Value = "+ b);
        System.out.println("short Value = "+ s);
        System.out.println("int Value = "+ v);
        System.out.println("int second Value = "+ calc);
        System.out.println("long Value = "+ amountVal);
        System.out.println("float Value = "+ intrestRate);
        System.out.println("double Value = "+ sineVal);
        System.out.println("boolean Value = "+ flag);
        System.out.println("boolean Value = "+ val);
        System.out.println("char Value = "+ ch1);
        System.out.println("char Value = "+ ch2);
    }
}
```

Output of above PrimitiveDemo class as below:



```
<terminated> PrimitiveDemo [Java Application] C:\Program Files\Java\jre6\bin\ja
byte Value = 100
short Value = 123
int Value = 123543
int second Value = -9876345
long Value = 1234567891
float Value = 12.25
double Value = 12345.234
boolean Value = true
boolean Value = false
char Value = X
char Value = Y
```

Summary of Data Types

Primitive Type	Size	Minimum Value	Maximum Value	Wrapper Type
char	16-bit	Unicode 0	Unicode 216-1	Character
byte	8-bit	-128	+127	Byte
short	16-bit	-215 (-32,768)	+215-1 (32,767)	Short
int	32-bit	-231 (-2,147,483,648)	+231-1 (2,147,483,647)	Integer
long	64-bit	-263 (-9,223,372,036,854,775,808)	+263-1 (9,223,372,036,854,775,807)	Long
float	32-bit	Approx range 1.4e-045 to 3.4e+038		Float
double	64-bit	Approx range 4.9e-324 to 1.8e+308		Double
boolean	1-bit	true or false		Boolean

Summary

- Java has group of variable types called primitive data type which are not object.
- Primitive types are categorize as Integer, Floating point, characters and boolean.
- Primitive types help for better performance of application.

Source: http://www.w3resource.com/java-tutorial/java-premitive-data-type.php