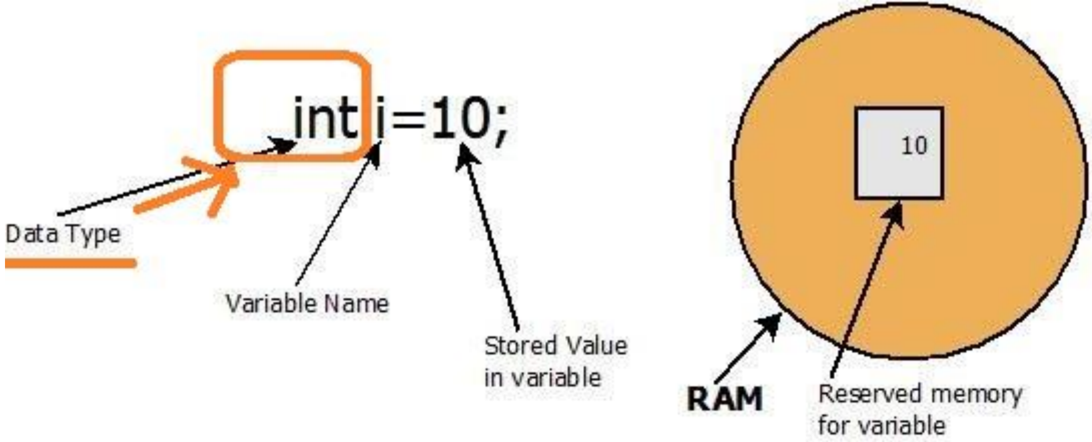
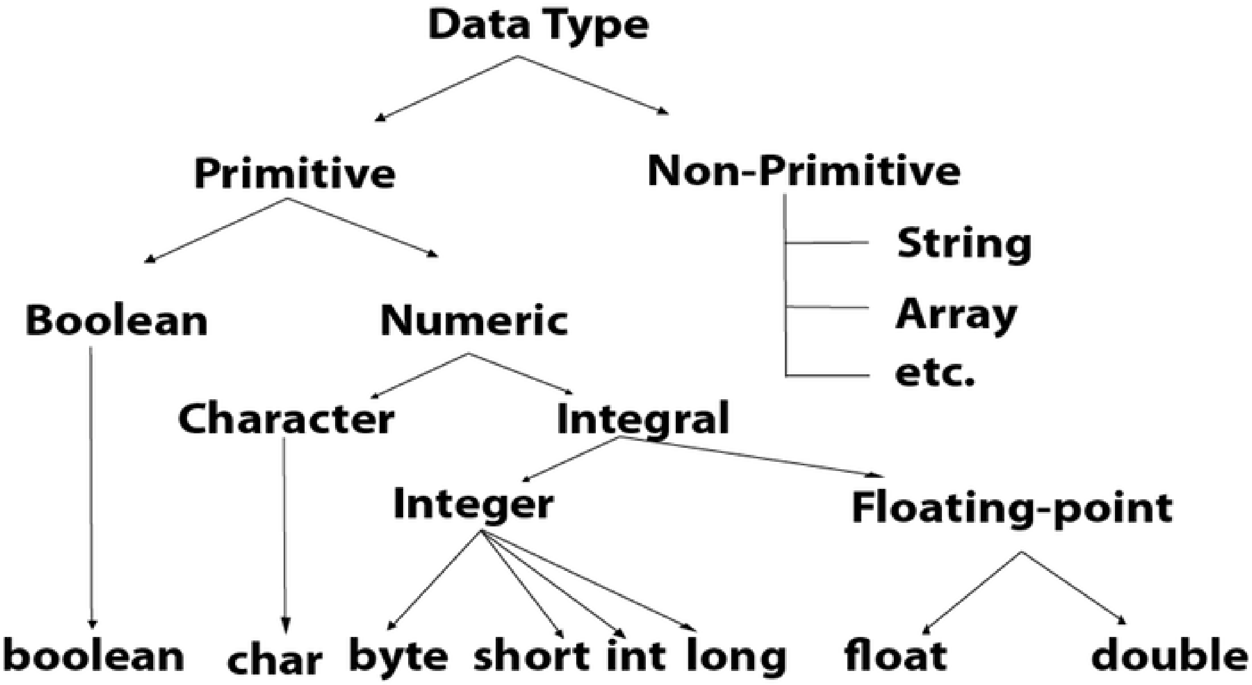
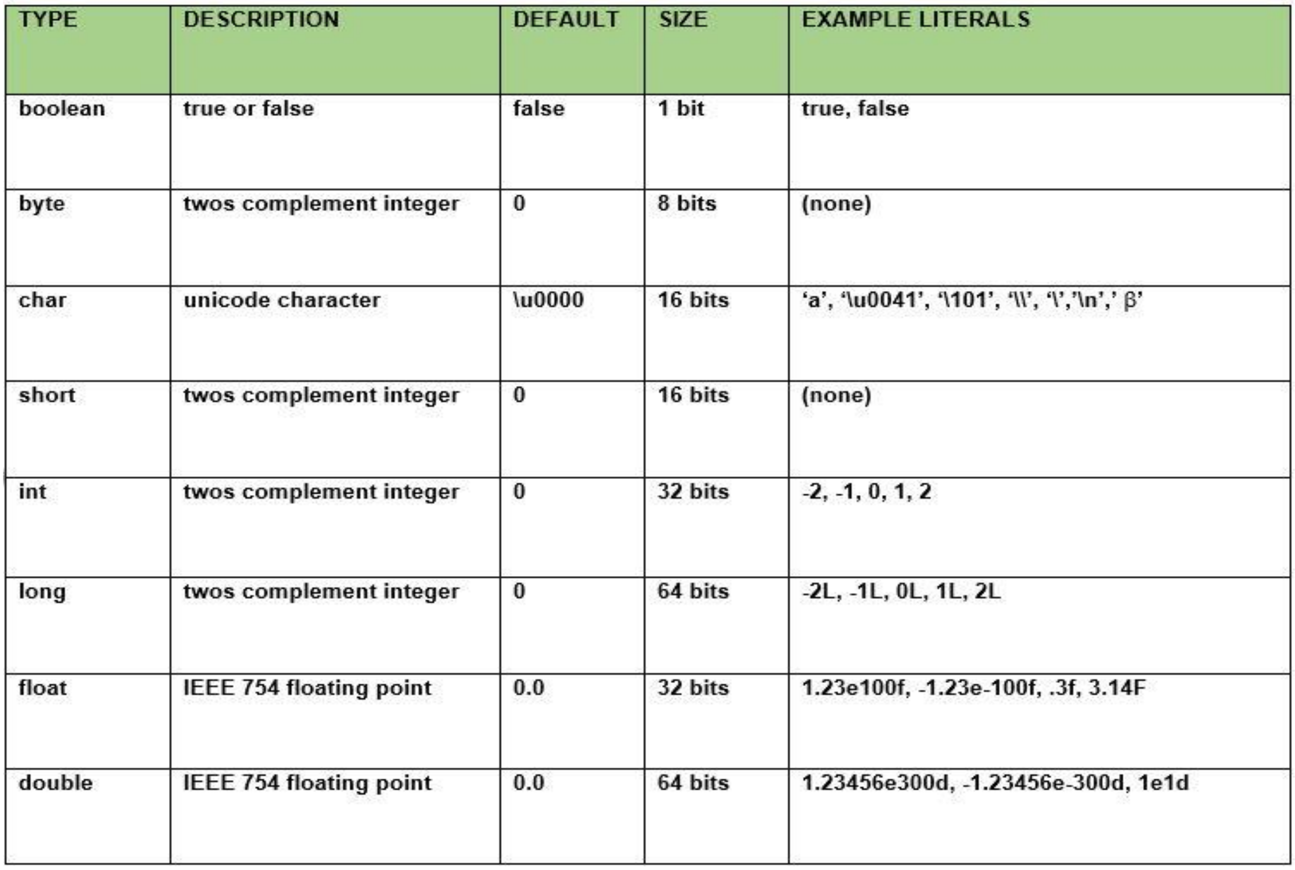
Lec _LAB2 2018-2019

Primitive types are the most basic **data types** available within the **Java** language. There are 8: boolean , byte , char , short , int , long , float and double . These **types** serve as the building blocks of **data** manipulation in **Java**. Such **types** serve only one purpose — containing pure, simple values of a kind.

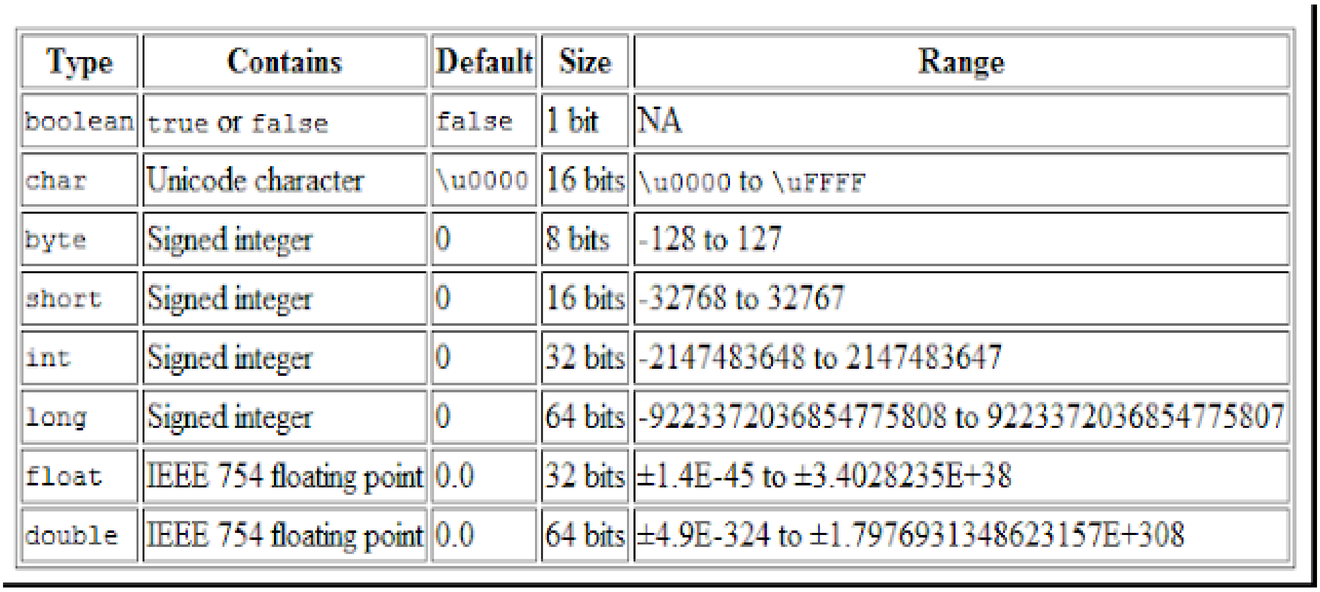




Lec _LAB2 2018-2019



TYPE	DESCRIPTION	DEFAULT	SIZE	EXAMPLE LITERALS
boolean	true or false	false	1 bit	true, false
byte	twos complement integer	0	8 bits	(none)
char	unicode character	\u0000	16 bits	‘a’, ‘\u0041’, ‘\101’, ‘\\’, ‘\’,’\n’,’ β’
short	twos complement integer	0	16 bits	(none)
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d



Type	Contains	Default	Size	Range
boolean	true or false	false	1 bit	NA
char	Unicode character	\u0000	16 bits	\u0000 to \uFFFF
byte	Signed integer	0	8 bits	-128 to 127
short	Signed integer	0	16 bits	-32768 to 32767
int	Signed integer	0	32 bits	-2147483648 to 2147483647
long	Signed integer	0	64 bits	-9223372036854775808 to 9223372036854775807
float	IEEE 754 floating point	0.0	32 bits	±1.4E-45 to ±3.4028235E+38
double	IEEE 754 floating point	0.0	64 bits	±4.9E-324 to ±1.7976931348623157E+308

Java Literals

A literal is a source code representation of a fixed value. They are represented directly in the code without any computation.

Literals can be assigned to any primitive type variable. For example −

```
byte a = 68;

char a = 'A';
```

byte, int, long, and short can be expressed in decimal(base 10), hexadecimal(base 16) or octal(base 8) number systems as well.

Prefix 0 is used to indicate octal, and prefix 0x indicates hexadecimal when using these number systems for literals. For example −

```
int decimal = 100;

int octal = 0144;

int hexa =  0x64;
```

```
boolean result = true;

  char capitalC = 'C';

  byte b = 100;

  short s = 10000;

  int i = 100000;
```

The integral types (byte, short, int, and long) can be expressed using decimal, octal, or hexadecimal number systems. Decimal is the number system you already use every day; it's based on 10 digits, numbered 0 through 9. The octal number system is base 8, consisting of the digits 0 through 7. The hexadecimal system is base 16, whose digits are the numbers 0 through 9 and the letters A through F. For general-purpose programming, the decimal system is likely to be the only number system you'll ever use. However, if you need octal or hexadecimal, the following example shows the correct syntax. The prefix 0 indicates octal, whereas 0x indicates hexadecimal.

```
int decVal = 26;  // The number 26, in decimal
int octVal = 032;         // The number 26, in octal
int hexVal = 0x1a;        // The number 26, in hexadecimal
```

The floating point types (float and double) can also be expressed using E or e (for scientific notation), F or f (32-bit float literal) and D or d (64-bit double literal; this is the default and by convention is omitted).

```
double d1 = 123.4;

double d2 = 1.234e2; // same value as d1, but in scientific notation

float f1  = 123.4f;
```