DATA TYPES

Agenda:

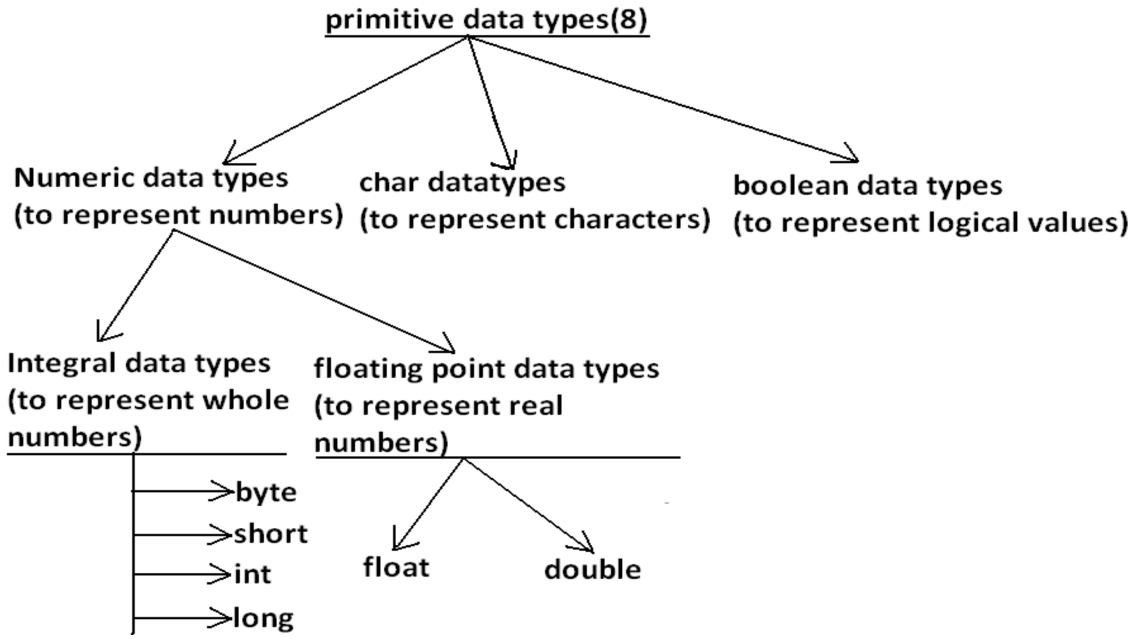
Data types

- Integral data types
    - byte
    - short
    - int
    - long
- floating point data types
- boolean data type
- char data type
- Java is pure object oriented programming or not ?
- Summary of java primitive data type

Data types:

Every variable has a type, every expression has a type and all types are strictly defined. More over every assignment should be checked by the compiler by the type compatibility Hence java language is considered as strongly typed programming language.

Diagram:

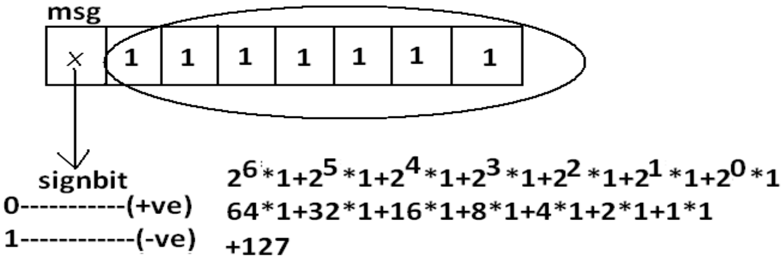


Except Boolean and char all remaining data types are considered as signed data types because we can represent both "+ve" and"-ve" numbers.

Integral data types :

byte:

```
Size: 1byte (8bits)
Maxvalue: +127
Minvalue:-128
Range:-128to 127[-2^7 to 2^7-1]
```



- The most significant bit acts as sign bit. "0" means "+ve" number and "1" means "–ve" number.
- "+ve" numbers will be represented directly in the memory whereas "–ve" numbers will be represented in 2's complement form.

```
Example:
byte b=10;
byte b2=130;//C.E:possible loss of precision
                     found : int
                     required : byte
byte b=10.5;//C.E:possible loss of precision
byte b=true;//C.E:incompatible types
byte b="ashok";//C.E:incompatible types
                         found : java.lang.String
                         required : byte
```

byte data type is best suitable if we are handling data in terms of streams either from the file or from the network.

short:

The most rarely used data type in java is short.

```
Size: 2 bytes
Range: -32768 to 32767(-2^15 to 2^15-1)
```

Example:

```
short s=130;
short s=32768;//C.E:possible loss of precision
short s=true;//C.E:incompatible types
```

short data type is best suitable for 16 bit processors like 8086 but these processors are completely outdated and hence the corresponding short data type is also out data type.

int:

This is most commonly used data type in java.

```
Size: 4 bytes
Range:-2147483648 to 2147483647 (-2^31 to 2^31-1)
```

Example:

```
int i=130;
int i=10.5;//C.E:possible loss of precision
int i=true;//C.E:incompatible types
```

long:

Whenever int is not enough to hold big values then we should go for long data type.

Example:

To hold the no. of characters present in a big file int may not enough hence the return type of length() method is long.

```
long l=f.length();//f is a file
Size: 8 bytes
Range:-2^63  to 2^63-1
```

Note: All the above data types (byte, short, int and long) can be used to represent whole numbers. If we want to represent real numbers then we should go for floating point data types.

Floating Point Data types:

float	double
If we want to 5 to 6 decimal places of accuracy then we should go for float.	If we want to 14 to 15 decimal places of accuracy then we should go for double.
Size:4 bytes.	Size:8 bytes.
Range:-3.4e38 to 3.4e38.	-1.7e308 to1.7e308.
float follows single precision.	double follows double precision.

boolean data type:

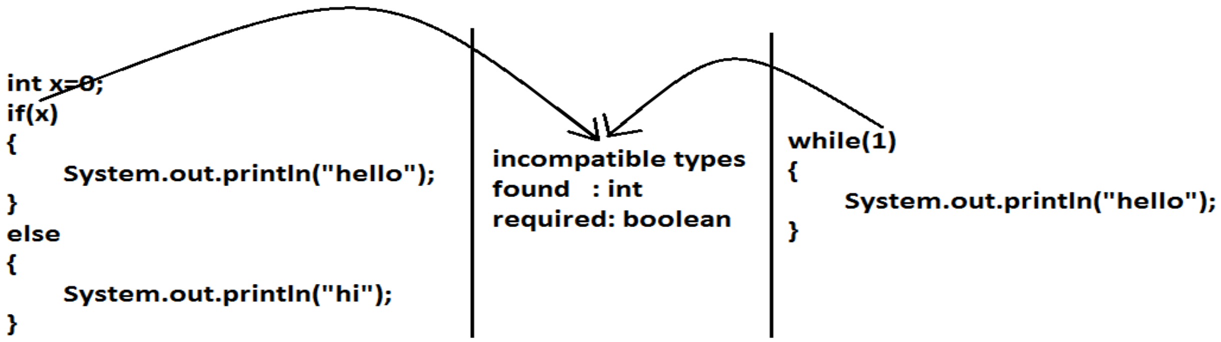
```
Size: Not applicable (virtual machine dependent)
Range: Not applicable but allowed values are true or false.
```

Which of the following boolean declarations are valid?

Example 1:

```
boolean b=true;
boolean b=True;//C.E:cannot find symbol
boolean b="True";//C.E:incompatible types
boolean b=0;//C.E:incompatible types
```

Example 2:



```
int x=0;
if(x)
{
	System.out.println("hello");
}
else
{
	System.out.println("hi");
}
```

```
incompatible types
found    : int
required: boolean
```

```
while(1)
{
	System.out.println("hello");
}
```

char data type:

In old languages like C & C++ are ASCII code based the no. of ASCII code characters are **< 256** to represent these 256 characters 8 - bits enough hence char size in old languages 1 byte.

In java we are allowed to use any worldwide alphabets character and java is Unicode based and no. of unicode characters are **> 256 and <= 65536** to represent all these characters one byte is not enough compulsory we should go for 2 bytes.

```
Size: 2 bytes
Range: 0 to 65535
```

Example:

```
char ch1=97;
char ch2=65536;//C.E:possible loss of precision
```

Java is pure object oriented programming or not?

Java is not considered as pure object oriented programming language because several oops features (like multiple inheritance, operator overloading) are not supported by java.

Moreover we are depending on primitive data types which are non objects.

Summary of java primitive data type:

data type	Size	Range	Corresponding Wrapper class	Default value
byte	1 byte	-2^7 to 2^7-1(-128 to 127)	Byte	0
short	2 bytes	-2^{15} to 2^{15}-1 (-32768 to 32767)	Short	0

int	4 bytes	-2^{31} to $2^{31}-1$ (-2147483648 to 2147483647)	Integer	0
long	8 bytes	-2^{63} to $2^{63}-1$	Long	0
float	4 bytes	-3.4e38 to 3.4e38	Float	0.0
double	8 bytes	-1.7e308 to 1.7e308	Double	0.0
boolean	Not applicable	Not applicable(but allowed values true|false)	Boolean	false
char	2 bytes	0 to 65535	Character	0(represents blank space)

The default value for the object references is "null".