Data Types

**Every variable has a type, every expression has a type and all the types are strictly defined and moreover every assignment should be checked by the compiler for the type compatibility hence java language is considered to be as strongly typed programming language.**

**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.**

**Eg 1:**

**To hold the distance travelled by light in 1000 days , int may not enough, compulsory we should go for long data type.**

**long l = 186000\*60\*24\*1000 miles**

**Eg 2:**

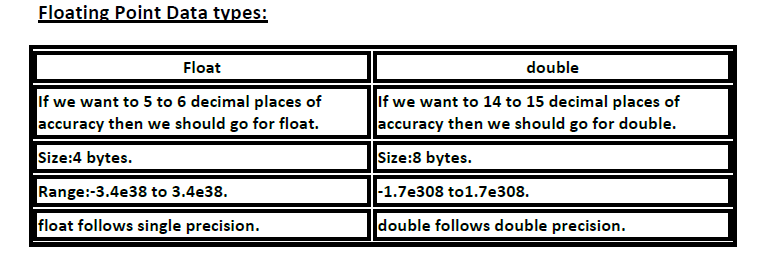
**To hold the number of characters present in a big file, int may not enough, compulsory we should go for long data type. Hence the return type of length() method is long.**

**long l=f.length();//f is a file**

**Size: 8 bytes**

**Range:-263 to 263-1**

**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.**



**boolean data type:**

**Size: Not applicable (virtual machine dependent)**

**Range: Not applicable but allowed values are true or false.**

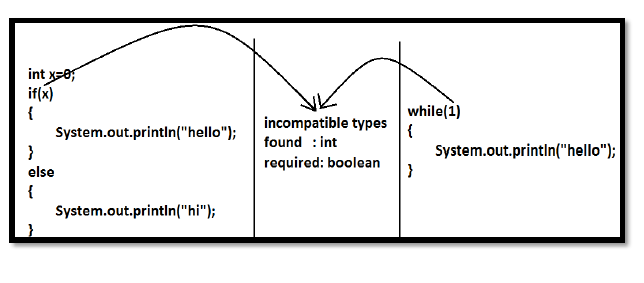
**Q. Which of the following boolean declarations are valid?**

**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**



**char data type:**

**Old languages like C & C++ are ASCII code based and the number of ASCII characters are < 256. To represent these 256 characters, 8 bits are enough and hence char size in old languages 1 byte.**

**But, in java we are allowed to use worldwide any alphabet character and java is Unicode based. The number of unicode characters are > 256 and <= 65536. To represent all these characters one byte is not enough compulsory we should go for 2 bytes.**

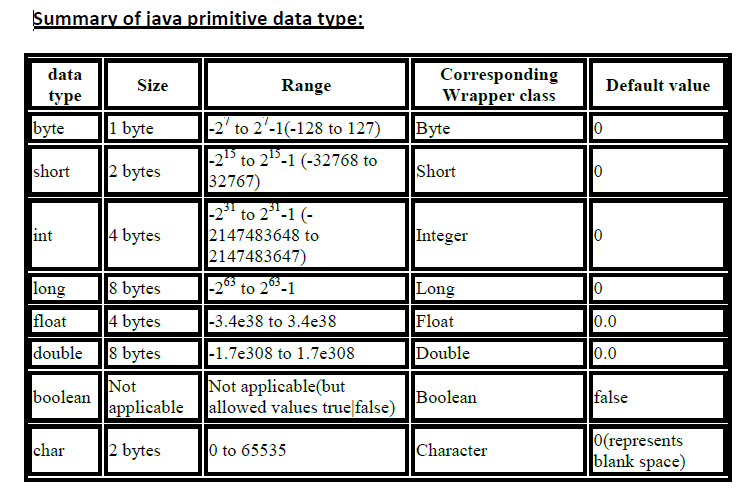
**Size: 2 bytes**

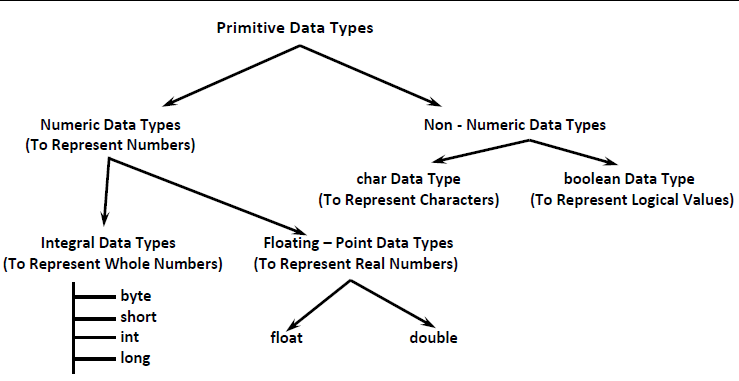
**Range: 0 to 65535**

**Example:**

**char ch1=97;**

**char ch2=65536;//C.E:possible loss of precision**





**Integral Data Types:**

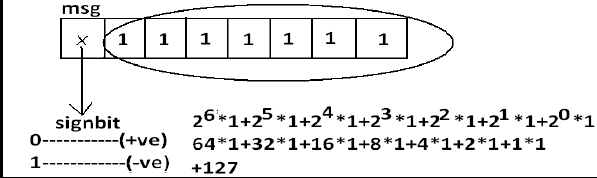
**byte Data Type:**

**Size: 1byte (8bits)**

**Maxvalue: +127**

**Minvalue:-128**

**Range:-128 to 127**



**Example:**

**byte b=10; // compiled with no error**

**byte b2=130;//Compilation Error:possible loss of precision**

**found : int**

**required : byte**

**byte b=10.5;//Compilation Error:possible loss of precision**

**byte b=true;//Compilation Error:incompatible types**

**byte b="ashok";//Compilation Error:incompatible types**

**found : java.lang.String**

**required : byte**

**short Data Type:**

**The most rarely used data type in java is short.**

**Size: 2 bytes**

**Range is-32768 to 32767(-215 to 215-1)**

**Example:**

**short s=130;**

**short s=32768;//C.E:possible loss of precision**

**short s=true;//C.E:incompatible types**

**Note:**

**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 outdated data type.**

**int Data Type:**

**This is most commonly used data type in java.**

**Size: 4 bytes**

**Range:-2147483648 to 2147483647 (-231 to 231-1)**

**Example:**

**int i=130;**

**int i=10.5;//C.E:possible loss of precision**

**int i=true;//C.E:incompatible types**

**long Data Type:**

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