Data types

In the declaration of every variable, it is mandatory to represent its data type.

syntax :

data\_type variable\_name ;

Example :

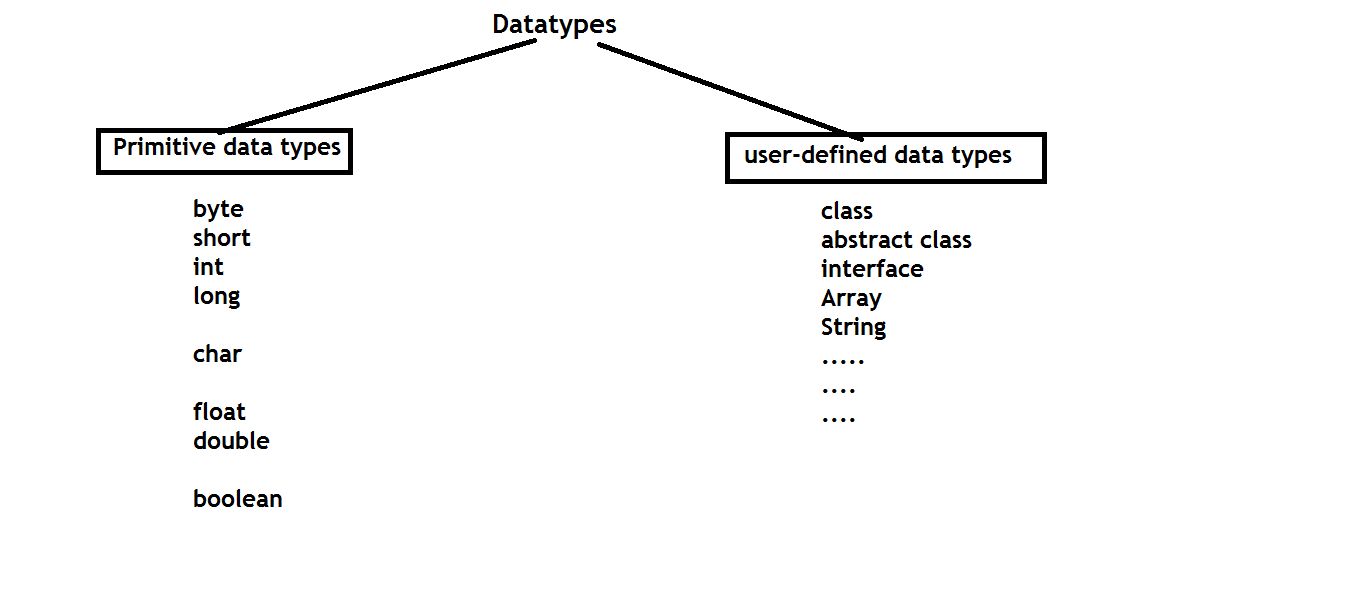
int sum = 0 ;

Why data type ?

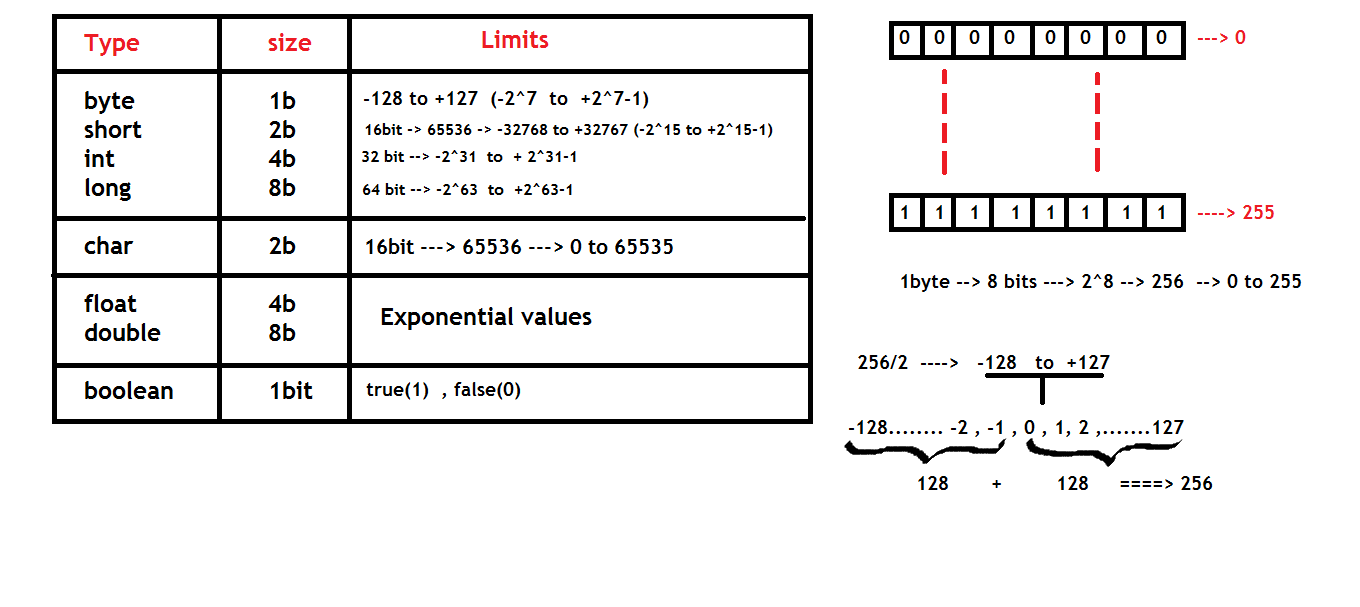
1) Describes how much memory is required to hold the data.

2) Describes what type of data is allowed.

Java is having pre-defined data type as well user-defined



Limits of each data type :



Questions on character data type :

1) why char data type limits discussion in integers ?

2) why character occupy 2 bytes memory in java and .net where as it occupies only 1 byte in C and C++?

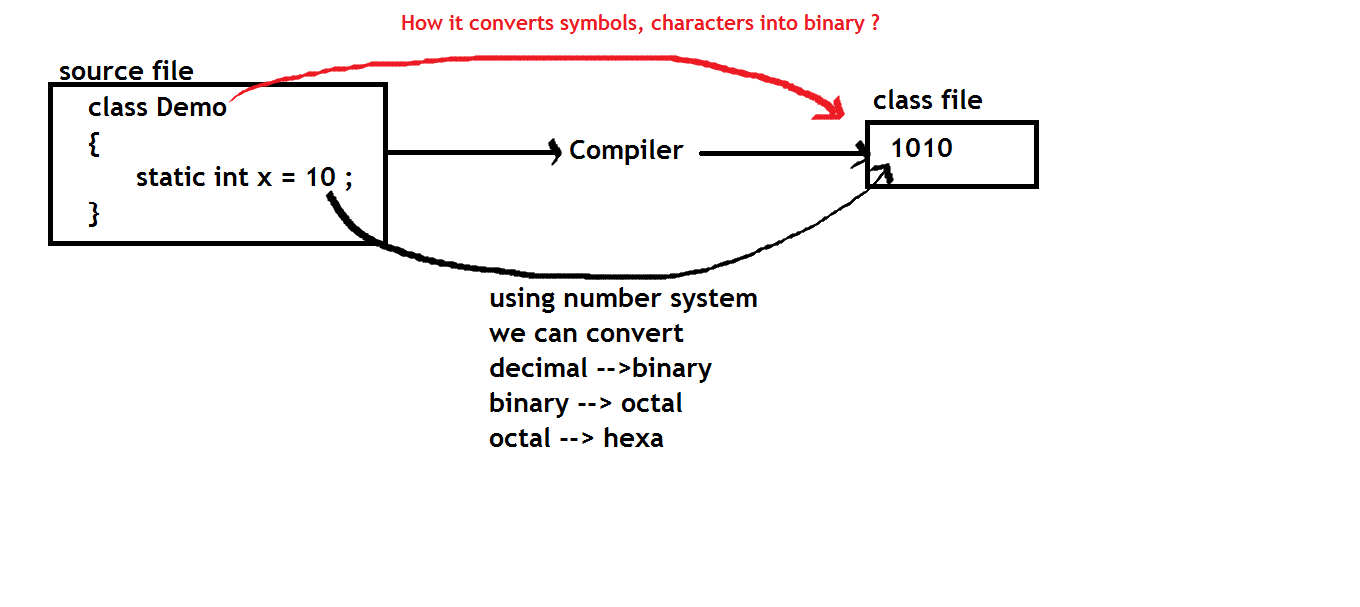
3) how can we store a symbol into 1 byte ?

4) what is character system ?

5) what is ASCII ?

6) what is UNICODE ?

Generally, compiler need to convert all the symbols of java source file into class file.



To convert all the characters of 1 language into machine code, we use character system.

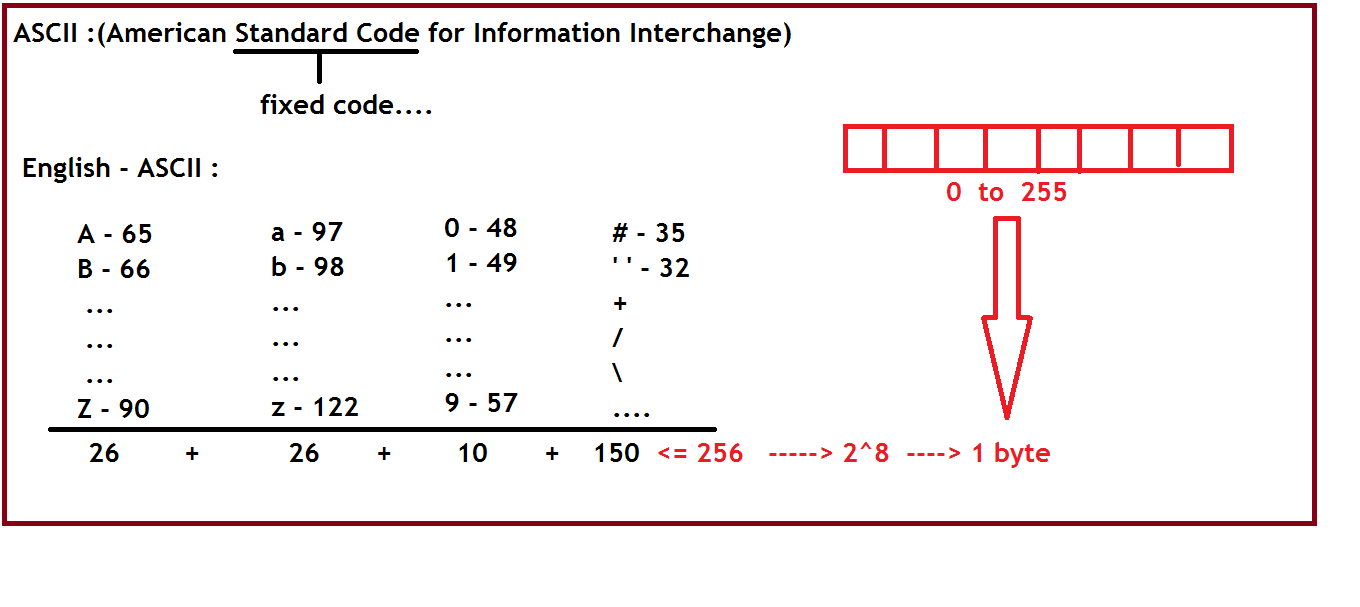
Introduced along with computer languages.

Compiler use character system for this conversion.

we have so many character systems.

Popular one is "ASCII"

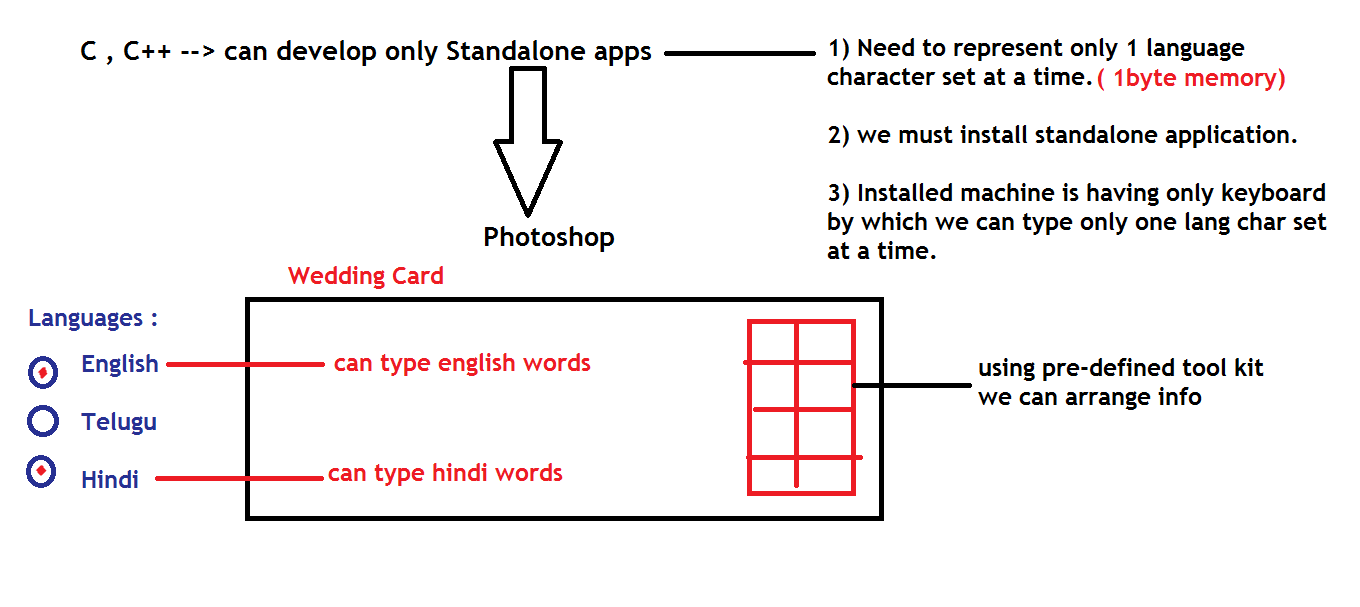
Using ASCII character set, we can represent all the symbols of one language using constant integer values.

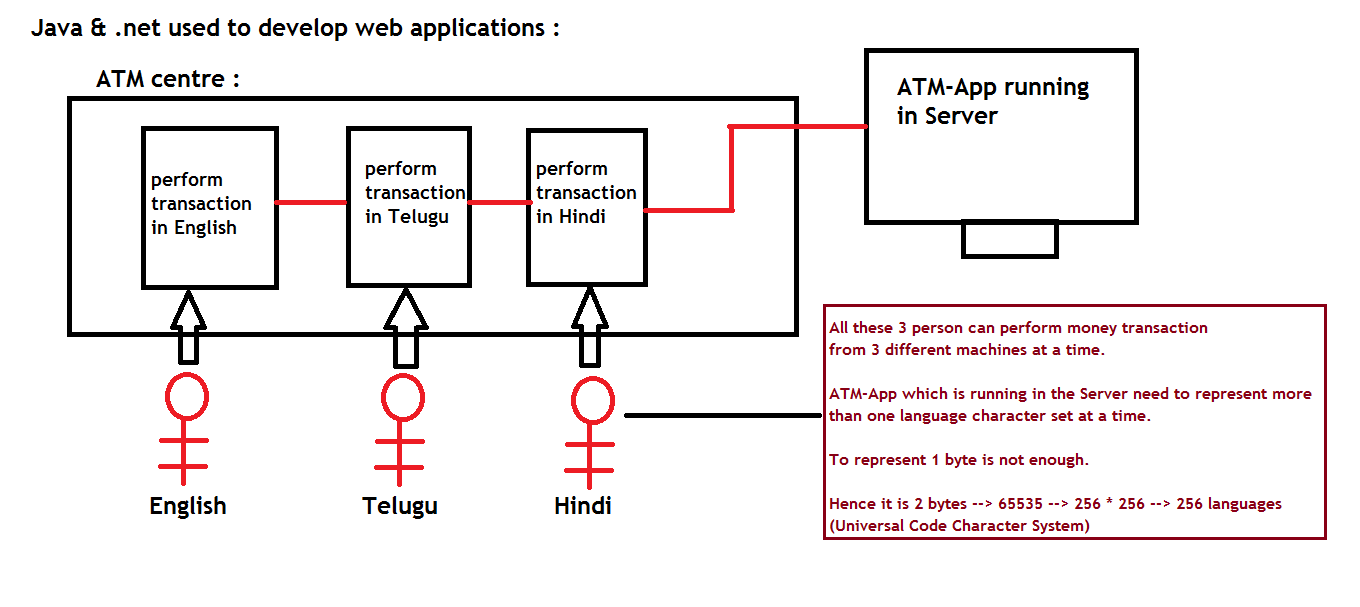


The best example that describes any language in this world is having not more than 256 symbols.

🡪 keyboard.

ASCII v/s UNICODE :





class DataTypes

{

public static void main(String[] args)

{

byte b = 100 ; //byte range -128 to +127

System.out.println("b value is : "+b); //string + value = concat

}

}

class DataTypes

{

public static void main(String[] args)

{

//byte b = 150 ; //CE : out of range

short s = 150 ; //Allowed (-32768 to +32767)

short s1 = 50000 ; //CE : out of range

}

}

class DataTypes

{

public static void main(String[] args)

{

byte a=10 , b=20 ;

//byte c = a+b ; //CE : arithmetic operations return result in integer format.

//short s = a+b ; //CE : not allowed

int c = a+b ; //Allowed

}

}

Type casting :

Converting data from one type to another at runtime.

Java allows

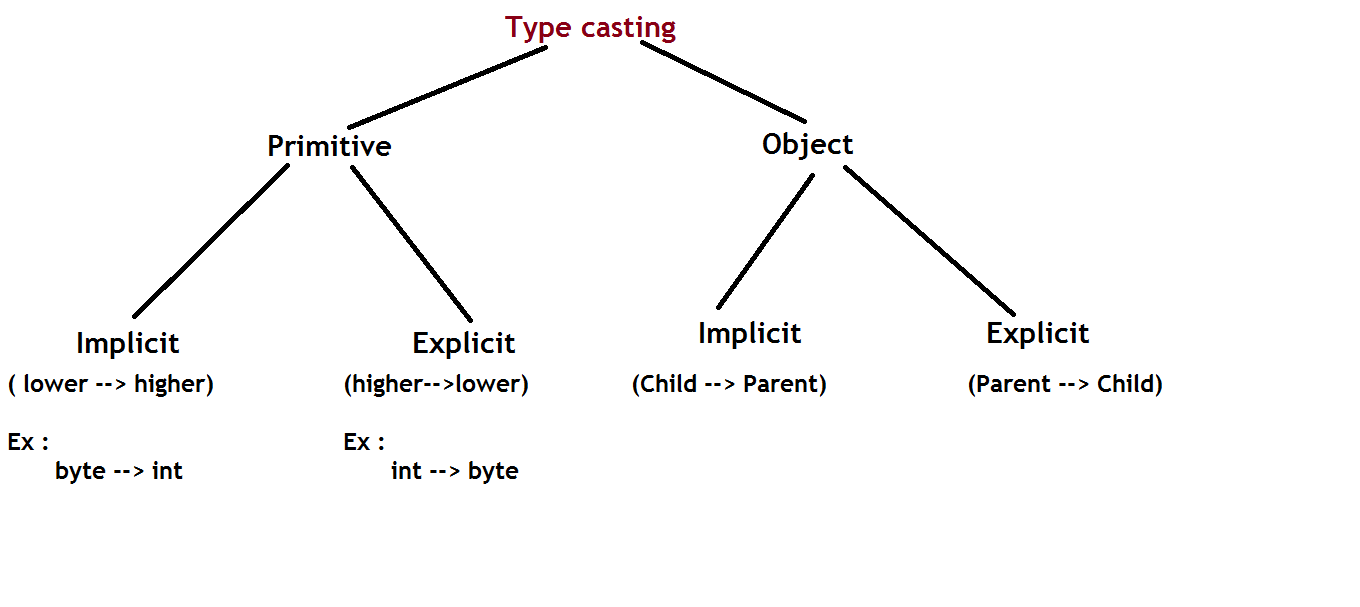
1) Primitive casting

2) Object casting

Casting can be either

1) implicit (internal) by JVM or

2) explicit (external) by Programmer.



class ImplicitCast

{

public static void main(String[] args)

{

byte b = 100 ;

int i ;

i = b ; //implicit cast.....

System.out.println("i value : "+i);

char ch = 'A' ;

int j ;

j = ch ; //implicit cast.....

System.out.println("j value : "+j);

}

}

class ExplicitCast

{

public static void main(String[] args)

{

int i = 100 ;

byte b ;

//b = i ; //direct assignment is not allowed

b = (byte)i ;

System.out.println("b value : "+b);

int j = 97 ;

char ch = (char)j ;

System.out.println("ch value : "+ch);

}

}

class ExplicitCast

{

public static void main(String[] args)

{

int i = 130 ;

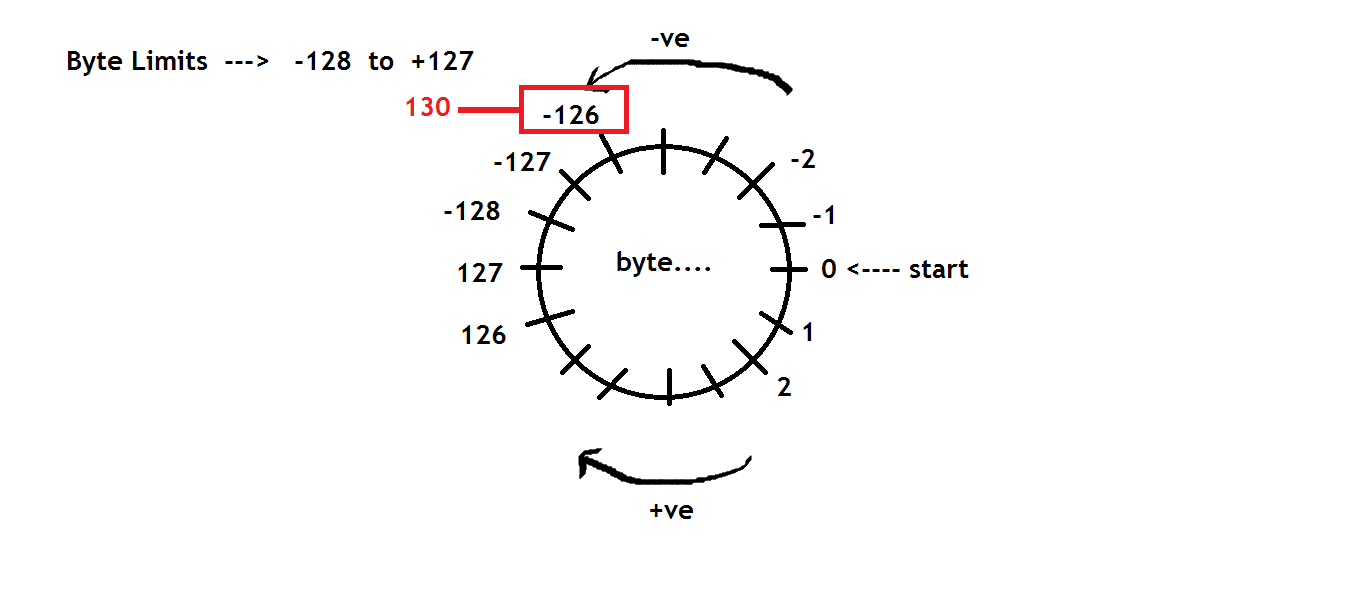
byte b ;

b = (byte)i ;

System.out.println("b value : "+b);

}

}



class ExplicitCast

{

public static void main(String[] args)

{

int i1 = 256 , i2 = 512 , i3 =1024 ;

byte b1, b2, b3 ;

b1 = (byte)i1 ;

b2 = (byte)i2 ;

b3 = (byte)i3 ;

System.out.println("b1 value : "+b1);

System.out.println("b2 value : "+b2);

System.out.println("b3 value : "+b3);

}

}

class ExplicitCast

{

public static void main(String[] args)

{

int i1 = -131 ;

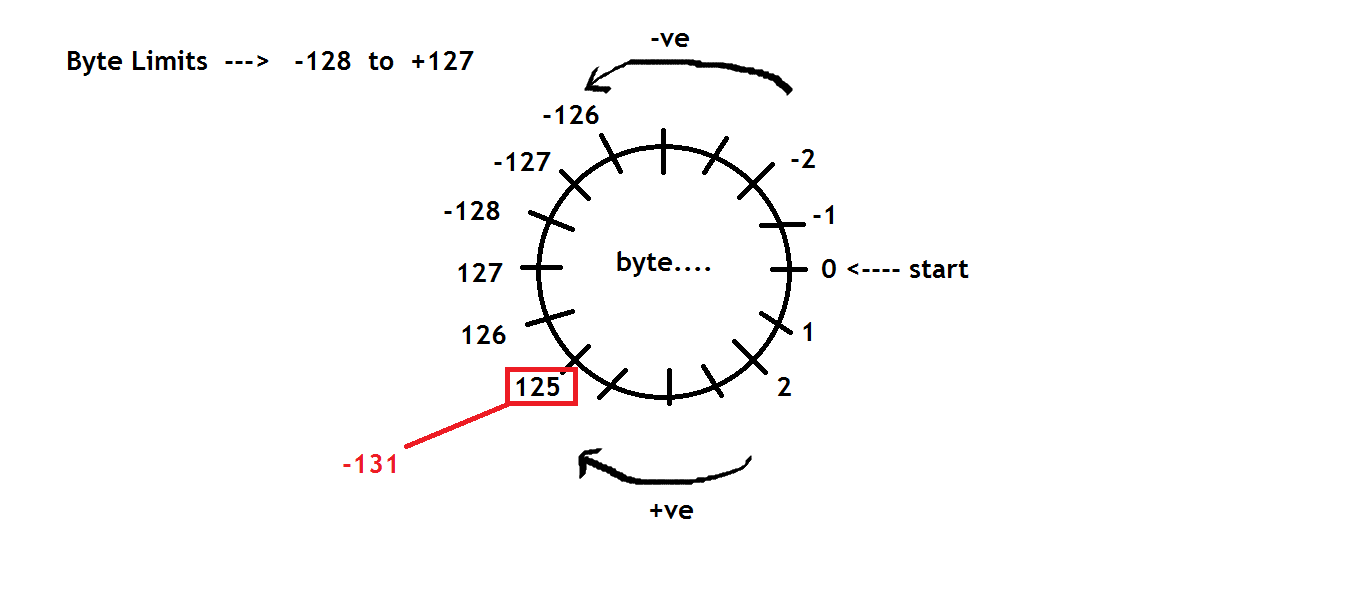
byte b1 ;

b1 = (byte)i1 ;

System.out.println("b1 value : "+b1);

}

}



Note : using type casting, we can conserve/save the memory in the application….

class DataTypes

{

public static void main(String[] args)

{

byte a=10 , b=20 ;

//int c = a+b ; //but occupies 4 byte memory.....

byte c = (byte)(a+b) ; //using typecast, if we store into byte variable, we can save 3 bytes memory.....

System.out.println("c value : "+c);

}

}

/\*

int / int --> int

int / float --> float

float / int--> float

float / float --> float

\*/

class DataTypes

{

public static void main(String[] args)

{

int x=5 , y=2 ;

int z = x/y ;

System.out.println("z value : "+z);

}

}

class DataTypes

{

public static void main(String[] args)

{

int x=5 , y=2 ;

float z = x/y ;

System.out.println("z value : "+z);

}

}

class DataTypes

{

public static void main(String[] args)

{

int x=5 , y=2 ;

float z = (float)x/y ;

System.out.println("z value : "+z);

}

}

class DataTypes

{

public static void main(String[] args)

{

int x=5 , y=2 ;

float z = (float)x/(float)y ;

System.out.println("z value : "+z);

}

}