**WRAPPER CLASSES**

Every primitive data type has its own wrapper class in java

WRAPPER CLASSES:

1.Byte

2.Short

3.Integer

4.Long

5.Float

6.Double

7.Charecter

8.Boolean

**Definition:**

To convert primitive data type into an object, we use Byte, Short, Integer, Long, Float, Double, Character, Boolean.

Since they wrap around primitives it is called **wrapper class.**

Converting primitive data type into an object using wrapper calss is called as **auto boxing**. (Compiler does it)

Converting an object to primitive data type is called as **un boxing**.(Developer should do)

In both cases we use wrapper class. For every data type, respective wrapper class is given by JDK

Every wrapper class will have overloaded constructor

1. Wrapper\_classname(data type)
2. Wrapper\_classname(String type)- here string should contain data type value

Example:

Integer i1= **new** ~~Integer~~(10); //allowed

Integer i2= **new** ~~Integer~~("1234");//allowed

Integer i3= **new** ~~Integer~~("raisha");// not allowed

for Integer Wrapper class, we should provide number format integer value(“1234”), if we pass other than number format(“raisha”), throws exception “java.lang.NumberFormatException”

(Except Character class).

**Object class**

* **Object class** is the Super most class of wrapper class.
* Object class is **supermost class and independent class, does not inherit any other class**
* Every class defined in java **inherits Object** class by default.

Program:

public class Demo91 {

Public static void main(String[] args) {

int i=10;

System.out.println(i);

Integer intObj1=new Integer(i);

System.out.println(intObj1);

**// boxing operation**

//similar data type calue in wrapper class constructor

Integer intObj2=new Integer(100);

**//string data type value in number format passed in wrapper class constructor**

Integer intObj3=new Integer("200");

//converstion of primitive data type to object type automatically is **autoboxing**.

Integer intObj4=300;

//converstion of primitive data type to object class type automatically **autocasting and** **autoboxing**.

Object intObj5=500;

**autocasting and** **autoboxing**.

Object ch=’r’;

}

}

**NOTE:BENEFITS OF AUTOBOXING AND AUTOCASTING**

**For generalisation of code,** auto boxing and auto casting is required.

EXAMPLE:

void display(Integer i)

{

sysout(i);

}

void display(FLOAT f)

{

sysout(f);

}

void display(Character c)

{

sysout(c);

}

**GENERALIZATION OF CODE**

//can accept 10,10.6,’c’,”dsfds”

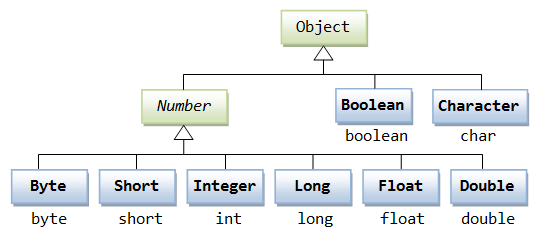
void display(Object o)

{

sysout(o);

}

WRAPPER CLASS HIERARCHY



**NOTE:**

Every number related wrapper class inherits from super class **Number** (**abstract class**).it is inherited from Object class. The Number class methods are over ridden in every wrapper class which deals with that number.

**UNBOXING**

Program:

public class Demo92 {

public static void main(String[] args) {

int i=100;

Integer intObj=new Integer(i);//boxing operation

System.out.println(intObj);

int j=intObj**;//unboxing operation**

System.out.println(j);

}

}

O/p:

100

100