**WRAPPER CLASSES:**

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 is called as **auto boxing**. (Compiler does it)

Converting an object(which is converted from auto boxing) 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 (Except Character class).

Object class is the Super most class of wrapper class.

**Note:**

Generally whenever a reference variable is printed the address of the object will be printed. The address is usually represented with fully qualified class name-fullyqualifiedclassname@hexadecimal address of the object.

Whenever a reference variable of any wrapper class is used or printed then it prints the primitive data of the wrapper class instead of address because in every wrapper class tostring() method of the Object class has been overridden to display or return the primitive data instead of address.

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

Integer intObj2=new Integer(100);

System.out.println(intObj2);

Integer intObj3=new Integer("200");

System.out.println(intObj3);

Integer intObj4=300;

System.out.println(intObj4);

}

}

o/p

10

10

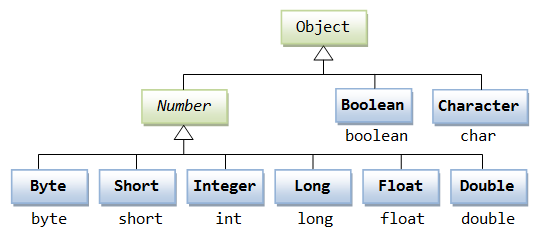
100

200

300

//Integer intObj2=100;√

//Integer intObj=i; √ (Auto Boxing) JDK 1.5 onwards



**Important points:**

1. Wrapper class belongs to java.lang package

2. In wrapper class tostring() method is over ridden

3. In wrapper class equals() method is over ridden

4. In wrapper class hashcode() method is over ridden

**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.

Program:

package com.qsp.pack2;

public class Demo92 {

public static void main(String[] args) {

// TODO Auto-generated method stub

int i=100;

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

System.out.println(intObj);

int j=intObj.intValue();//unboxing operation

System.out.println(j);

}

}

O/p:

100

100

Program:

package com.qsp.pack2;

public class Demo93 {

public static void main(String[] args) {

// TODO Auto-generated method stub

int i=120;

Integer intObj=new Integer(120);

System.out.println(intObj);

int j=intObj.intValue();

System.out.println(j);

double d=intObj.doubleValue();//unboxing and auto-widening

System.out.println(d);

//1st way

Double dd=new Double(d);//boxing

System.out.println(dd);

//2nd way

Double dd1=d;//boxing

System.out.println(dd1);

//unboxing

double d2=dd1.doubleValue();

int i3 =dd1.intValue();//unboxing and narrowing

int i4 =(int)dd1.doubleValue();//unboxing and explicit narrowing

byte b=123;

Long l1=new Long(b);//b widening

System.out.println(l1);

short s=125;

Long l2=new Long(s);//s widening

System.out.println(l2);

double d4=12.80;

Long l3=new Long((long)d4);//explicit narrowing,double is greater than long

System.out.println(l3);

long l4=100;

Byte b1=new Byte((byte)l4);//Explicit narrowing,long is greater than byte

System.out.println(b1);

}

}

O/P:

120

120

120.0

120.0

120.0

123

125

12

100

Converting an Object to string

we use toString() method to convert an object into string,look at the program to understand better

Program;

package com.qsp.pack2;

public class Demo94 {

public static void main(String[] args) {

// TODO Auto-generated method stub

Double d1=new Double(120.0);

System.out.println(d1.toString());

Integer i1=100;

System.out.println(i1.toString());

Long l=1255l;

System.out.println(l.toString());

Double dd=new Double(12.05);

String s=dd.toString();

System.out.println(s);

//converting primitive to string

//there are 2 toString()methods 1.non static and 2.static

//non static is accessed by creating an object, just like above

//static can be accessed by using classname.toString(),just like below and within braces we can pass the value

Double.toString(12.05);

Long.toString(100l);

Byte.toString((byte)100);

//String s1=l, cannot be converted from long to string assigning l address to s1

//adding two strings containing numbers and put it in another string

String s3="123";

String s4="456";

String s5=Integer.toString(Integer.parseInt(s3)+Integer.parseInt(s4));

//we use Integer.parseInt();to convert string into int.

System.out.println(s5);

Integer i4=new Integer(123);

Integer i5=i4;//we can assign like this also

System.out.println(i4==i5);

System.out.println(i5);

Integer i6=Integer.valueOf(i5);//returning integer object

System.out.println(i6==i5);

}

}

O/P:

120.0

100

1255

12.05

579

true

123

false

Program:

package com.qsp.pack2;

class C{

void test(int i)

{

System.out.println("int method is implemented");

}

void test(double d)

{

System.out.println("double method is implemented");

}

void test(Number num)

{

System.out.println("Number method is implemented");

}

void test(Integer ii)

{

System.out.println("Interger method is implemented");

}

void test(Object o)

{

System.out.println("object method is implemented");

}

}

public class Demo95 {

public static void main(String[] args) {

// TODO Auto-generated method stub

C c1=new C();

c1.test(10);

}

}

o/p:

int method is implemented

Execution Preferences

Preference of execution->Same type, Auto widening, Wrapper classes, Number class and Object class

i.e

1.int

2.double

3.Integer

4.Number

5.Object