As the name suggests, wrapper classes are **objects encapsulating primitive Java types**. Each Java primitive has a corresponding wrapper.

Each of java primitive data types has a class and those classes are called wrapper classes because they wrap the data into an object.

|  |  |
| --- | --- |
| **Primitive Data Type** | **Wrapper Class** |
| Byte | Byte |
| Short | Short |
| Int | Integer |
| long | Long |
| float | Float |
| double | Double |
| boolean | Boolean |
| Char | Character |

All wrapper classes are in java.lang.\* package.

**Convert int to Integer:**

Int x = 10;

Integer y = new Integer(x);

**Convert Integer to int:**

Integer a = new Integer(10);

Int x = a.intValue();

**public** **static** **void** main(String[] args)

{

System.***out***.println("Convert int value to Integer.");

**int** x = 10;

Integer a = **new** ~~Integer~~(x);

System.***out***.print(a);

System.***out***.println("\n\nConvert Integer to int");

Integer c = **new** ~~Integer~~(20);

**int** b = c.intValue();

System.***out***.print(b);

System.***out***.println("\n\nAutoBoxing -> Converting premitive type to the corresponding reference type.");

**int** n = 100;

Integer m = n;

System.***out***.print(m);

System.***out***.println("\n\nAuto UN-Boxing -> Converting reference type to the corresponding premitive type.");

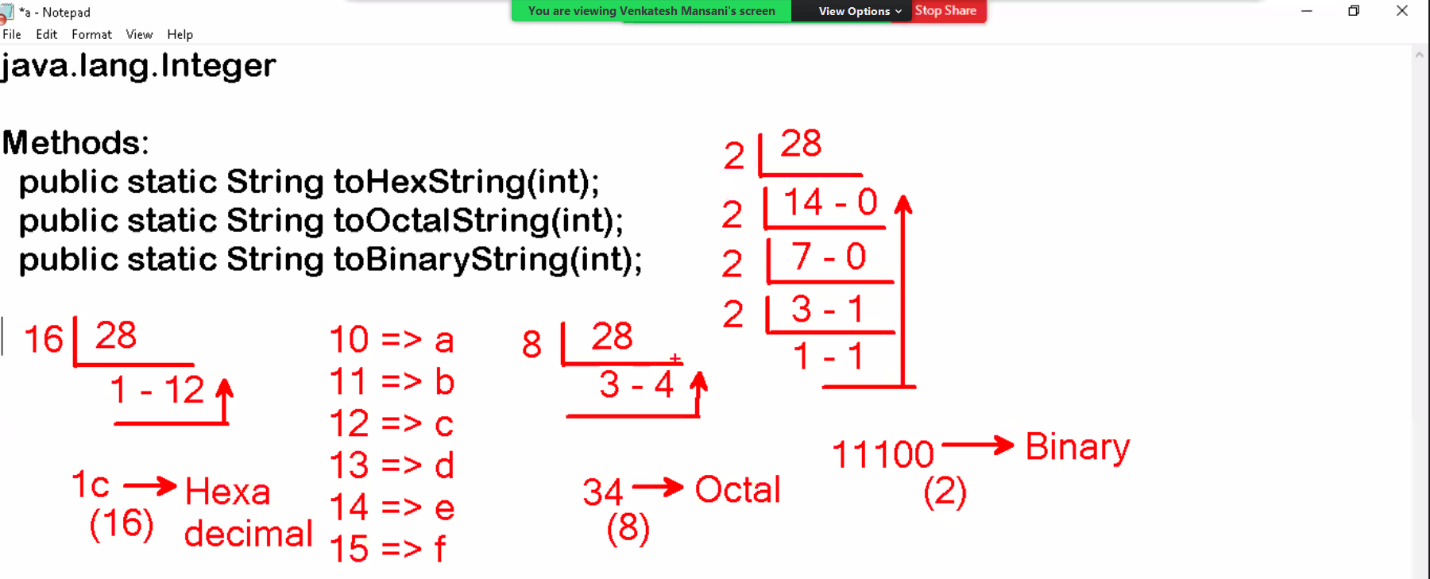
Integer x1 = 200;

**int** y = x1;

System.***out***.print(x1);

}

Refer code snippet: WrapperClassExample1\_Class



**public** **class** Demo {

**public** **static** **void** main (String args[])

{

**int** x = 28;

System.***out***.println(x);

String s1 = Integer.*toHexString*(x);

System.***out***.println(s1);

String s2 = Integer.*toOctalString*(x);

System.***out***.println(s2);

String s3 = Integer.*toBinaryString*(x);

System.***out***.println(s3);

}

}

//output

28

1c

34

11100

Converting Binary to Decimal, Octal to Decimal, Hexadecimal to decimal => no method required.

But converting Decimal to Octal, Decimal to Hexadecimal , Decimal to Binary => method required.

**public** **class** Demo {

**public** **static** **void** main (String args[])

{

**int** x = 0b11100; // Binary literal, Binary literals are prefixed wit 0b or 0B

System.***out***.println(x); // Octal literals, they are prefixed with 0;

**int** y = 034; // Hexadecimal Literals, they are prefixed with 0x or 0X

System.***out***.println(y);

**int** z = 0x1c;

System.***out***.println(z);

}

}

//output

28

28

28

Method:

Public static int parseint(String) throws NumberFormatException;

* It is used to parse int from given String.

Static imports:

This feature allows to avoid class name with static members.

Syntax:

Import static package-name.sub-package-name.ClassName.\*;

Note: Here \* indicates all static members.

Example:

Import static java.lang.integers.\*;

This static allows to access all static members of integer class without class name.

**import** **static** java.lang.Integer.\*;

**import** **static** java.lang.System.\*;

**public** **class** Demo {

**public** **static** **void** main (String args[])

{

**int** x = *parseInt*("210");

**int** y = *parseInt*("345");

***out***.print(x+y);

}

}

//output

555