Wrapper Class:

Wrapper classes help us to convert primitive types to object types.
We can convert Primitive to Object is called as Boxing.
We can convert Object to Primitive is called as Unboxing.

Types:
Wrapper: Boolean, Byte, Character, Double, Float, Integer, Long, Short
Primitive: boolean, byte, char, double, float, int, long, short

Wrapper Class:

Primitive Type to Object Type (3 Ways)

1.Using Parameterized Constructor
2.Using valueOf() method
3.Auto boxing

Object Type to Primitive Type (2 Ways)

1.Using intValue() method
2.Auto Unboxing

primitive type to object type 3 ways:

```
//using parameterized constructor
int i = 10;
System.out.println("Primitive Type: " + i);
Integer integer = new ~~Integer(i); // The constructor Integer(int) has been deprecated since version 9 and marked for removal~~
System.out.println("Object Type: " + integer);

//converting primitive type to object type using valueOf() method from wrapper classes
int j = 10;
System.out.println("Primitive Type: " + j);
Integer valueOf = Integer.valueOf(j);
System.out.println("Object Type: " + valueOf);

//converting primitive type to object type using auto boxing
int k = 10;
System.out.println("Primitive Type: " + k);
Integer auto = k;
System.out.println("Object Type: " + auto);
```

object type to primitive type 2 ways:

```
//converting  object type to primitive type using intValue() method
Integer i = new ~~Integer(10);~~
System.out.println("Object type: " + i);
int intValue = i.intValue();
System.out.println("Primitive type: " + intValue);

// converting  object type to primitive type using auto unboxing
Integer i1 = new ~~Integer(10);~~
System.out.println("Object type: " + i1);
int i2 = i1;
System.out.println("Primitive type: " + i2);
```

Wrapper Class:

String Type to Object Type (2 Ways)

1.Using Parameterized Constructor

2.Using valueOf() method

Object Type to String Type (2 Ways)

1.Using toString() method

2.Using + Operator

string type to object type 2 ways:

```
//converting  string type to object type using parameterized constructor
String s1 = "10";
System.out.println("String type: " + s1);
Integer i = new ~~Integer(s1);~~
System.out.println("Object type: " + i);

//converting  string type to object type using valueOf() method
String s2 = "10";
System.out.println("String type: " + s2);
Integer i2 = Integer.valueOf(s2);
System.out.println("Object type: " + i2);
```

object type to string type 2 ways:

```
// converting object type to string type using toString()
Integer integer = new ~~Integer(10);~~
System.out.println("Object type: " + integer);
String s1 = integer.toString();
System.out.println("String type: " + s1);

// converting object type to string type using + operator
Integer i1 = new ~~Integer(10);~~
System.out.println("Object type: " + i1);
String s2 = "" + i1; //"" + i1.toString()
System.out.println("String type: " + s2);
```

Wrapper Class:

Primitive Type to String Type (2 Ways)

1.Using toString method

2.Using + Operator

String Type to Primitive Type (1 Way)

1.Using paresInt() method

primitive type to string type 2 ways:

```
// converting primitive type to string type using toString()
int i = 10;
System.out.println("Primitive type: " + i);
String s1 = Integer.toString(i);
System.out.println("String type: " + s1);

//converting object type to string type using + operator
int i1 = 10;
System.out.println("Primitive type: " + i1);
String s2 = ""+i1;
System.out.println("String type: " + s2);
```

string type to primitive type 1 way:

```
String s1 = "10";
int parseInt = Integer.parseInt(s1);
System.out.println(parseInt);
```