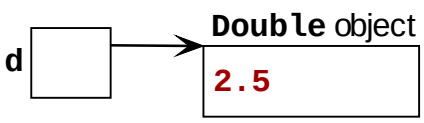
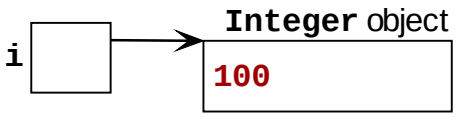
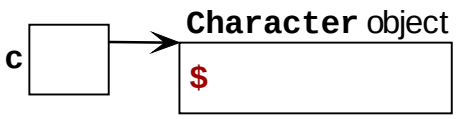
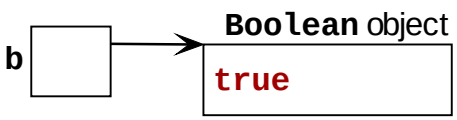
WRAPPER CLASSES

Each primitive data type in Java has a corresponding wrapper class in package **`java.lang`**. A ***wrapper class*** is used to wrap a primitive datum within an object.

Java Primitive Data Type	Corresponding Wrapper Class in java.lang
`byte`	`Byte`
`short`	`Short`
`int`	`Integer`
`long`	`Long`
`char`	`Character`
`boolean`	`Boolean`
`float`	`Float`
`double`	`Double`

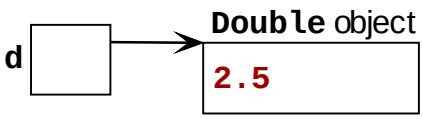
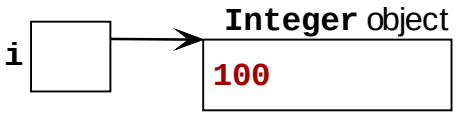
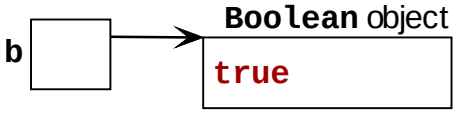
Objects of a wrapper class can be built using a constructor that expects an argument of the corresponding primitive data type.

Examples

`Double d = new Double( 2.5 );`	
`Integer i = new Integer( 100 );`	
`Character c = new Character( '$' );`	
`Boolean b = new Boolean( true );`	

Furthermore, each wrapper class except **`Character`** has an overloaded constructor that accepts a string argument.

Examples

`Double d = new Double( "2.5" );`	
`Integer i = new Integer( "100" );`	
`Boolean b = new Boolean( "true" );`	

If the string argument to one of these constructors is not a valid representation of the primitive datum, then the constructor will throw an exception.

Example

This Java statement:

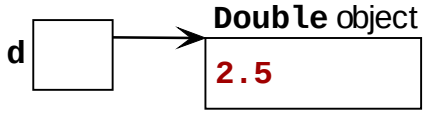
```
Integer i = new Integer( "2.5" );
```

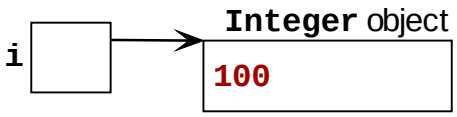
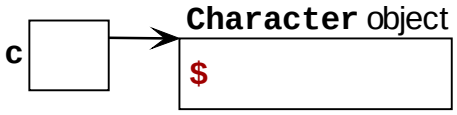
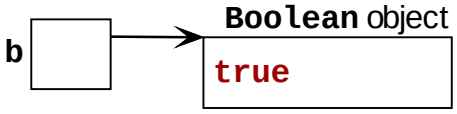
Throws the run-time error:

```
Exception in thread "main" java.lang.NumberFormatException
```

If you use a primitive datum where an object is required, the Java compiler automatically replaces it by an object of the corresponding wrapper class. This is called ***autoboxing*** the primitive.

Examples

`Double d = 2.5;`	

`Integer i = 100;`	
`Character c = '$';`	
`Boolean b = true;`	

If you use a wrapper class object where a primitive datum is expected, the Java compiler automatically ***unboxes*** the object by replacing it with the primitive datum.

Example

In line 1 below, the compiler automatically boxes the primitive **`100`** into the **`Integer`** object **`i`**. In line 2 it unboxes **`i`** in order to apply the **`/`** operator. The code outputs the value **`50`**.

```
1  Integer i = 100;
2  System.out.println( i/2 );
```

Exercises

1.	Write the Java statement that converts **`1E6`** to a **`Double`** object.
2.	Write the Java statement that converts **`1_000_000`** to an **`Integer`** object.
3.	Write the Java statement that converts **`true`** to a **`Boolean`** object.
4.	Write the Java statement that converts **`'A'`** to a **`Character`** object.