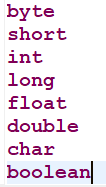
**Wrapper classes**

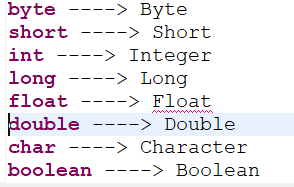
Java is not 100% object oriented because, we have concept of primitive data types which are not objects.

We have 8 primitive data types in Java and 53 keywords. Below is the list of primitive data types.



To make a java program 100% object oriented, we make use of wrapper class concept to convert the primitive data types to objects.

Below is list of wrapper classes corresponding to each primitive data type.



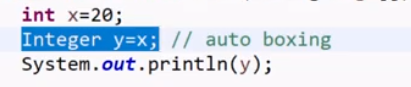
**Note:** Using wrapper class concept, we can convert the primitive data types to objects or from objects to primitive data types.

**Auto boxing**

Converting the primitive data type to object is taken care by java default in the upgraded versions.

This automatic process of converting the primitive data type to object is called as **“Auto boxing”.**

**Below is the example.**



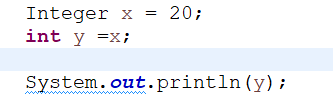
Here x value is directly assigned to y, which is a wrapper class object.

Internally wrapper class would get the value of this primitive data type and store that to y.

Ex:



**Converting object to primitive data type**



Here also auto-boxing concept is used for converting object to primitive data type.

Internal process looks like below.



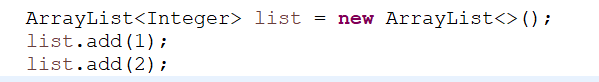
But this is not required from java 9. As auto boxing will take care of this.

**Note:** One more use of wrapper classes is in collections.

Collection is nothing but group of objects but not primitive datatypes. So, only objects can be stored.

In such cases, we have to make use of wrapper classes and not datatypes.

Ex:



As we can see here, while providing the collection type, we have mentioned Integer but not int.

While adding, though we are directly adding int value, which will be converted as object using auto-boxing concept.