

[Return to Module Overview Page \(https://onlinelearning.berkeley.edu/courses/665040/pages/module-six\)](https://onlinelearning.berkeley.edu/courses/665040/pages/module-six)

---

## 6.3: Autoboxing

JDK 1.5 introduced autoboxing (also known as boxing conversions). Java will automatically convert a primitive to its corresponding wrapper type when necessary. It also introduced auto unboxing, which goes the other way (converting a wrapper type to its corresponding primitive data type).

Both autoboxing and auto unboxing make code more readable and less error prone.

The following example demonstrates the proper way to work with both primitives and object wrappers without autoboxing:

```
Integer intObj = new Integer(100);
int intPrim = 1;
int resultPrim = intObj.intValue() + intPrim;
Integer resultObj = new Integer(intObj.intValue() + intPrim);
```

Essentially, all we want to do is add the values of 1 and 100, but as you can see there is plenty of extra, but necessary code.

With autoboxing, the code above can now be rewritten in the following way:

```
Integer intObj = new Integer(100);
int intPrim = 1;
int resultPrim = intObj + intPrim;
Integer resultObj = intObj + intPrim;
```

Notice how much clearer and more readable the revised code is. In addition, you do not lose the benefits of compile time type checking.