

16.4 Autoboxing and Auto-Unboxing

Java provides boxing and unboxing conversions that automatically convert between primitive-type values and type-wrapper objects. A **boxing conversion** converts a value of a primitive type to an object of the corresponding type-wrapper class. An **unboxing conversion** converts an object of a type-wrapper class to a value of the corresponding primitive type. These conversions—called **autoboxing** and **auto-unboxing**—are performed automatically. Consider the following statements:

```
Integer[] integerArray = new Integer[5]; // create int
integerArray[0] = 10; // assign Integer 10 to integer
int value = integerArray[0]; // get int value of Inte
```



In this case, autoboxing occurs when assigning an `int` value (`10`) to `integerArray[0]`, because `integerArray` stores references to `Integer` objects, not `int` values. Auto-unboxing occurs when assigning `integerArray[0]` to `int` variable `value`, because variable `value` stores an `int` value, not a reference to an `Integer` object. Boxing conversions also occur in conditions, which can evaluate to primitive `boolean` values or `Boolean` objects. Many of the examples in [Chapters 16–21](#) use these conversions to store

primitive values in and retrieve them from data structures.