

Solutions to Big Java Chapter 6 Review Exercises

6.1, 6.2

R6.8

False. For `A && B` if `A` is false, `B` is never evaluated. Consider the following.

```
n != 0 && (x / n == 3)
```

R6.12

The `==` operator tests for equality of object references. The `equals` method tests for equality of object contents.

```
String one = "Mi$$i$$ippi";
String two = "Mississippi".replace("s", "$");
String three = one;
System.out.println(one == two);
System.out.println(one.equals(two));
System.out.println(one == three);
```

Prints

```
false
true
true
```

R6.13

As in R6.12, the `==` operator tests whether 2 `Rectangle` references refer to the same object. The `equals` method tests whether 2 `Rectangle` objects contain the same contents (starting coordinates, width, and height).

R6.14

It is not possible to run any method, including `equals`, on a reference that refers to `null`. If `r` refers to `null`, the code throws a `NullPointerException` when `equals` is run on `r`. The correct check for `null` is `r == null`.

6.3, 6.4

R6.3

Let `a` and `b` be initialized `boolean` variables. The text printed at each point in the code below is accurate.

```
// if/else/else
if(a)
    System.out.println("a is true.");
else if(b)
    System.out.println("a is false. b is true.");
else
    System.out.println("Both a and b are false.");

// nested if
if(a)
{
    if(b)
        System.out.println("Both a and b are true.");
}
else
    System.out.println("a is false. b is unchecked.");
```

R6.4

Let `grade` be an initialized `int` variable such that $0 \leq \text{grade} \leq 100$. The order of the tests in the following code does not matter.

```
if(grade >= 90)
    System.out.println("A");
else if(80 <= grade && grade < 90)
    System.out.println("B");
else if(70 <= grade && grade < 80)
    System.out.println("C");
else if(65 <= grade && grade < 70)
    System.out.println("D");
else if(grade < 65)
    System.out.println("F");
```

The code can be modified to use fewer comparisons. In the modified code below, the order of the tests matters.

```
if(grade >= 90)
    System.out.println("A");
else if(grade >= 80)
    System.out.println("B");
else if(grade >= 70)
    System.out.println("C");
else if(grade >= 65)
    System.out.println("D");
else
    System.out.println("F");
```

R6.9

In the first case, both tests are always run. At the end of the first code segment, *s* could have the values 0, 1, or 2. In the second case, the test with *y* is only run if the test with *x* is false. At the end of the second code segment, *s* could have the values 0 or 1.