

## Lab 8: Boolean Values and Conditional Computation

### **Objectives:**

1. To explore logical (Boolean) expressions.
2. To use the Java conditional statements `if` and `if-else`.
3. To write a method that returns a boolean value.

### **Preparation:**

1. Go over the Lecture Notes: Topic 8 pages, Topic 10.
2. Textbook Reading (optional): Chapter 6, Sections 6.1 and 6.2.

### **Exercise 1: Evaluating logical expressions**

In DrJava's Interactions Pane, type in the following variable declarations and initializations:

```
int a = 12;
int b = 7;
boolean t = true;
char c = 'e';
```

Now replace the question mark in each the following logical (also called “Boolean”) expressions with one of the logical operators `&&` `||` `^` or one of the relational operators `==` `!=` `<` `>` as you type them in to the Interactions pane, so that each of the expressions evaluates to `true`. (Note that there may be more than one way to complete each expression correctly.)

```
(a > 0) && (b ? 0)

('a' > c) ? ('E' != c)

(a > b) ^ (t ? (c > 's'))

(a % b != 0) ? (b * 2 > a)

(t ? !(a > b)) && t
```

### **Exercise 2: Using an if statement in a while loop**

In Lab 7 Exercise 3, you completed a program `EnterCorrectInput` that wanted a user to enter a number between 1 and 100, and kept prompting the user until the input was in the correct range. There is another common way of checking whether or not an input is in the correct range, using a boolean variable and an if statement to indicate whether or not the loop should continue. Study the code provided below; it is a variation of `EnterCorrectInput`. The boolean variable `cont` is set to `false` when it is determined that

the loop should **not** continue. (Note: we might have wanted to use the identifier `continue` for the boolean variable here, but that is not allowed because `continue` is a keyword in Java.) Fill in the blank space in the program below so that it functions the same as the program of Lab 7. (Hint: the condition in the if statement should now check if the input is in the *correct* range.) Test your program for boundary values, as you did in Lab 7 Exercise 3 b).

```
public class EnterCorrectInput2
{
    public static void main(String[] args)
    {
        int num;
        boolean cont = true;    // continue if cont is true
        System.out.println("Enter a number between 1 and 100.");

        while (cont)

        {
            num = SimpleInput.getIntNumber("Enter number: ");

            if ( _____ )
            {
                cont = false;

                System.out.println("The number you entered is " + num);
            }
        }
    }
}
```

### Exercise 3: Using an if statement in a Picture method

1. Starting with the `copyPictureTo` method you used in Lab 5 (and Assignment 2), change it to a new method called `copyExceptWhite` that will copy all but the white pixels from the source picture to the target picture.
2. Use this method to put the robot in *robot.jpg* and/or the turtle in *turtle.jpg* on the moon in *moon-surface.jpg*. (These files are all in the `mediasources` folder).

### Exercise 4: Using an if-else statement in a Picture method

1. Create a new method called `blueExtremes( )` in the `Picture` class that removes blue from every pixel that has a blue value less than 128, and otherwise gives the pixel a maximum blue value.
2. Test this method from the Interactions pane, using the image in *grayMotorcycle.jpg*.