

CS 157 Lab 10 November 3, 2009

while loops

1. Create a new class called **Lab10**. Write a static method named `showTwos` that shows the factors of 2 in a given integer. For example, the following calls produce the following output:

<u>Call</u>	<u>Output</u>
<code>showTwos(7);</code>	<code>7 = 7</code>
<code>showTwos(18);</code>	<code>18 = 2 * 9</code>
<code>showTwos(68);</code>	<code>68 = 2 * 2 * 17</code>
<code>showTwos(120);</code>	<code>120 = 2 * 2 * 2 * 15</code>

You are to express the number as a product of factors of 2 and an odd number. The number 120, for example, has 3 factors of 2 multiplied by the odd number 15. The idea is that as long as the number you're working with is even, you can print a 2 and divide the number by 2. For odd numbers (as in the first example of 7), there are no factors of 2, so you just show the number itself. Assume that your method is passed a number greater than 0.

Test your code with the following:

```
showTwos(7); // 7 = 7
showTwos(18); // 18 = 2 * 9
showTwos(68); // 68 = 2 * 2 * 17
showTwos(120); // 120 = 2 * 2 * 2 * 15
showTwos(32); // 32 = 2 * 2 * 2 * 2 * 2 * 1
```

2. Write a static method named `getNames` that repeatedly prompts the user for names. Once the user types three asterisks ("***"), all names typed are displayed. For example:

```
Type a person's name (or *** to stop): Marty
Type a person's name (or *** to stop): Andrea
Type a person's name (or *** to stop): Stuart
Type a person's name (or *** to stop): ***
Welcome to all: Marty Andrea Stuart
```

3. Write a static method named `getNumbers` that repeatedly prompts the user for a nonnegative integer (or a negative number to stop). When the user enters a negative number, the method prints the largest and the smallest numbers entered. For example:

```
Type a number (or -1 to stop): 4
Type a number (or -1 to stop): 11
Type a number (or -1 to stop): 24
Type a number (or -1 to stop): 2
Type a number (or -1 to stop): 3
Type a number (or -1 to stop): -1
The smallest number was 2 and the largest number was 24
```

4. Write a static method named `toBinary` that takes as a parameter an integer in base 10 and returns the integer equivalent in base 2. Here is how to convert a base 10 number to binary. If we call the base 10 number n then the remainder upon dividing n by 2 is the units digit of the binary number. Now divide n by 2. The remainder of that number upon division by 2 is the next digit to the left in the binary number. The easiest thing to do is to concatenate those computed digits into a String. Repeat this process as long as you're not dealing with zero. Here are some base 10 numbers and their binary equivalents:

Base 10 number	Binary number
5	101
32	100000
31	11111
10	1010
0	0

Create JUnit tests for this method. Feel free to add other cases as well.

Copy both `Lab10.java` and `Lab10Test.java` to the shared drive.

3. Write a program that prompts the user for two students' overall scores and that computes their letter grades based on those scores. Assume that scores are in the range of 0 to 100 and that grades are based on the following scale:

Score	Grade
< 60	0.0
60-62	0.7
63	0.8
64	0.9
65	1.0
...	
93	3.8
94	3.9
>= 95	4.0

You should also report whether or not the students passed the course (i.e., whether or not the user got a grade above 0.0) and whether the student may advance to the next course (i.e., whether or not the user got a grade of 2.0 or higher). If the student's grade percent has a decimal, round it to the nearest whole number.

Here is a sample log of execution of the program (user input underlined):

```
This program converts two scores into grades.
```

```
Student 1:
```

```
What score did you get? 83.7
```

```
You received a grade of 2.8
```

```
You passed the course
```

```
You may advance to the next course
```

```
Student 2:
```

```
What score did you get? 59.1
```

```
You received a grade of 0.0
```

```
You failed the course
```

```
You must take the course again
```

Use static methods to capture the structure and redundancy of the program. Your program should have at least two static methods other than `main`.

Factoring if/else Code

4. In the following Java program, rewrite the `getCoordinates` method using factoring to eliminate redundancy. Your method must still produce the same output and result under all cases.

```
import java.util.*;

public class Coordinates {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        double result = getCoordinates(console);
        System.out.println(result);
    }

    public static double getCoordinates(Scanner console) {
        System.out.print("What is the x coordinate? ");
        double x = console.nextDouble();
        int negatives;
        if (x < 0.0) {
            System.out.print("What is the y coordinate? ");
            double y = console.nextDouble();
            if (y < 0.0) {
                negatives = 2;
                System.out.println("negatives = " + negatives);
                return x + y + negatives;
            } else {
                negatives = 1;
                System.out.println("negatives = " + negatives);
                return x + y + negatives;
            }
        } else {
            System.out.print("What is the y coordinate? ");
            double y = console.nextDouble();
            if (y < 0.0) {
                negatives = 1;
                System.out.println("negatives = " + negatives);
                return x + y + negatives;
            } else {
                negatives = 0;
                System.out.println("negatives = " + negatives);
                return x + y + negatives;
            }
        }
    }
}
```

5. The code in the following Java program is redundant, basically doing the same thing twice. Write a static method named `getBills` that eliminates this redundancy. Rewrite method `main` to use your new method.

```
import java.util.*;

public class Bills {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        System.out.print("How much will John be spending? ");
        double amount = console.nextDouble();
        System.out.println();
        int numBills1 = (int) (amount / 20.0);
        if (numBills1 * 20.0 < amount) {
            numBills1++;
        }

        System.out.print("How much will Jane be spending? ");
        amount = console.nextDouble();
        System.out.println();
        int numBills2 = (int) (amount / 20.0);
        if (numBills2 * 20.0 < amount) {
            numBills2++;
        }

        System.out.println("John needs " + numBills1 + " bills");
        System.out.println("Jane needs " + numBills2 + " bills");
    }
}
```

Chapter 4 Lab Handout Solutions

1. Two possible solutions appear below.

```
public static int median(int x, int y, int z) {
    if (x < y) {
        if (z < x) {
            return x;
        } else if (y < z) {
            return y;
        } else { // x < z < y
            return z;
        }
    } else {
        if (z < y) {
            return y;
        } else if (x < z) {
            return x;
        } else { // y < z < x
            return z;
        }
    }
}
}
```

```
public static int median(int x, int y, int z) {
    return x + y + z -
        Math.min(x, Math.min(y, z)) -
        Math.max(x, Math.max(y, z));
}
```

2.

```
public static double tax(double salary) {
    if (salary <= 7150) {
        return 0.1 * salary;
    } else if (salary < 29050) {
        return 715 + 0.15 * (salary - 7150);
    } else if (salary < 70350) {
        return 4000 + 0.25 * (salary - 29050);
    } else { // salary >= 70350
        return 14325 + 0.28 * (salary - 70350);
    }
}
}
```

4.

```
public static double getCoordinates(Scanner console) {
    System.out.print("What is the x coordinate? ");
    double x = console.nextDouble();
    System.out.print("What is the y coordinate? ");
    double y = console.nextDouble();

    int negatives = 0;
    if (x < 0.0) {
        negatives++;
    }
    if (y < 0.0) {
        negatives++;
    }
    System.out.println("negatives = " + negatives);
    return x + y + negatives;
}
```

5.

```
public class Bills {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);

        int numBills1 = getBills(console, "John");
        int numBills2 = getBills(console, "Jane");

        System.out.println("John needs " + numBills1 + " bills");
        System.out.println("Jane needs " + numBills2 + " bills");
    }

    public static int getBills(Scanner console, String name) {
        System.out.print("How much will " + name + " be spending? ");
        double amount = console.nextDouble();
        System.out.println();

        int numBills = (int) (amount/20.0);
        if (numBills * 20.0 < amount) {
            numBills++;
        }

        return numBills;
    }
}
```