Lab 02: Back to the Future

CO225: Software Construction

June 21, 2016

1 Goals

By the end of this lab, you should be able to:

- Write a Java program with a main method
- Use System.out.println to send output to the console
- Use Scanner class for keyboard input
- Read both numeric and String data, taking care to avoid the pitfalls that may occur with newline characters

2 Introduction

A Java program is really a class definition with a method named main. In other words, when a Java program is run, the method named main is the first method that is invoked. Up until now, weve given you a default Main.java to include in your assignments. In this lab well write our own main method, and will also investigate keyboard input and console output.

2.1 Introduction to I/O(Input/Output) in Java

Console output

send some output to the console (intelliJ) window. Java pre-defines an object called System.out that provides a mechanism to send output (as a String) to the window. If you want the String "Bye Bye, cruel world" to appear when you run your program, you include the following statement in the main method:

```
// Main.java
public class Main {
    public static void main(String args[]) {
          System .out.println("Bye Bye, cruel world");
    }
}
```

Each call to System.out.println generates a new line of output. You can also provide the String to be output as a variable.

```
   String message = "Bye Bye, cruel world";
   System.out.prntln(message);
```

In fact, with println you can output the values of almost any kind of object or primitive. If you want to output more than one thing, place an addition sign between the things you want to output (+ is the concatenation operator for Strings):

```
  int day;
  day  = 17;
  System .out.println("Today is Wed, Nov " + day + ".");
```

Keyboard Input

Java includes a class called Scanner that makes it easy to read input from the keyboard. Import java.util.Scanner into your Main class. Java's name for the keyboard is System.in, so create a new Scanner object like this:

```
    Scanner   keyboard = new Scanner(System.in);
```

You can name your Scanner object anything you want, but programmers often name it keyboard when the input that will be read into the program comes from the keyboard. Now you can use the methods of the Scanner class with the object keyboard to input data that the user of the program types in.

```
    name = input.next();
```

Use the Scanner method nextInt() to read an integer value in from the keyboard

```
   int amount = keyboard.nextInt();
```

Multiply the amount variable with 2016. Use System.out.println() to display the result. Run your program. You may have been confused when you ran your program, because although the program was waiting for you to type in a number, it didnt tell you that. Whenever you do keyboard input (interactive input), you should precede each input operation with a prompt: a message that is displayed on the console that notifies the user what the program is expecting. Use System.out.println() to display a prompt, and run the program again. By the way, System.out.print() works just like System.out.println(), except that with println, the next output goes on a new line, whereas with print, the next output is placed on the same line (in other words,println outputs a newline after it outputs the String in its argument list). Prompt messages are often written using print rather than println.

Whitespace and Newlines

Youve probably noticed that you have to be very specific about the way you space your output when you use print or println. For example,

```
int day = 7;
System.out.println("Here’s today’s date:" + "June" + day + "," +
    "2016");
```

produces very different output from

```
int day = 7;
System.out.println("Here’s today’s date:" + "Nov " + day + ", "
    +" 2010");
```

(Try it and see.) Whitespace is any sequence of characters such as spaces, tabs, and newlines, that appear as white space when written on paper. Whitespace also acts as a separator when you read in multiple input items from the keyboard. Each activation of the Scanner method nextInt() skips over any intervening whitespace. So, for example, if your program reads values into two int variables num1 and num2:

```
int num1;
int num2;

num1 = keyboard.nextInt();
num2 = keyboard.nextInt();
```

The input may be entered at the keyboard in a number of ways:

- the two numbers may be entered on the same line, with any number of space and/or tab characters before and/or after each number
- the two numbers may be entered on two separate lines, one number on each line, with any number of space/tab characters before/after each number
- the two numbers may be entered on two separate lines with any number of blank lines before or in between

Scanner has a method called nextLine() that reads the remainder of a line of input starting wherever the previous reading left off, and returns the input read in as a String. nextLine() reads up to the next newline character (denoted in a Java program by '\n'...this is the character that is read when the user types the ENTER key). When nextLine() reads a line of input, it reads the newline character (so that the next reading of input will begin on the next line) but the '\n' character itself isnt returned as part of the String.

The fact that nextLine() reads the newline character but doesnt store it as part of the returned String can be the cause of confusion when both numeric and text input are being read for a newPrecinctobject this way (here were entering the pop first, followed by the preName, and finally the preAddr)

```
int pop = keyboard.nextInt();
String preName = keyboard.nextLine();
String preAddr = keyboard.nextLine();
```

If the user types in the input on three separate lines like this

```
1673
Holmes S
42, Baker Street
```

You might expect the value of pop to be 1673, the value of preName to be "Holmes S", and the value of preAddr to be "42, Baker Street". But what youll actually get is the value of pop is 1673, the value of preName is "" (the empty string), and the value of preAddr is "Holmes S" (put the code into your program and try it). Make sure you can explain why. Can you think of a way to modify the code so that the values are what we first expected? Modify this code to make it work.

Theres a lot more that can be said about keyboard input and console output. Java has a printf() method that is part of the PrintStream class that is used the same way as print and println but lets you format your output according to a format string argument. There are also classes such as DecimalFormat that can be used to format output in a particular way. The Scanner class contains many methods that we didnt discuss that can be used to read in various kinds of data. Start trying to read the Javadocs (Java Documentation, such as that provided for Scanner) to learn about additional I/O capabilities.

3 What to turn in?

Task 01

A number is special, if it is exactly divisible by 15. A number is big, if it is greater than 999. A number is weird, if it is exactly divisible by 5 and 6 but not 18. A number is scary if it is big or weird. Write a Java program to test a given number. For example:

```
Enter a number: 450
450 is special but not scary.
Enter a number: 750
750 is special, weird and scary but not big.
```

Task 02

When doing any work with visual media it is often very useful to have the complement of a color on hand to create contrast and bring the focus of a picture to a particular place. To create the complement of a color on a computer, each of the red, green, and blue values of a color are inverted. Each of the red, green, and blue values of a color can range from 0 to 255, inclusive. If a particular component of one color is 50, then that component of its complement is 255-50=205.

Although this generally works well, it doesn't generate good complements for grey colors that have all three components right around 128. To fix this you will return an alternate complement for grey colors. If each component of a color and its corresponding component of the color's complement differ by 32 or less, then make the complement of each component by either adding 128 to a component's value, or by subtracting 128 from a component's value, whichever one results in a legal value. For example, the color 115,115,143 would have the complement 140, 140,112, but since each component of the complement would have been within 32 of the corresponding component of rgb, we return the alternate complement 243, 243, 15 instead.

Write a programme that takes a color representing the red, green, and blue values of a color, in that order, and prints the red, green, and blue values of the complement of that color, in that same order. Following is a typical input/output scenario and the format:

```
Enter the color: 255 0 0
The complement: 0 255 255
```