Computer Science, Claremont McKenna College
CS51 - Introduction to Computer Science, Spring 2015

Problem Set 4 Due: 11:55 PM, Feb 22 2015

General Instructions

Please carefully read and follow the directions exactly for each problem. Files and classes should be named exactly as directed in the problem (including capitalization!) as this will help with grading.

You should create your programs using your preferred text-editor, the Eclipse text editor, or jGrasp. Do not use a word processor such as Microsoft Word, WordPad, Google Docs, Apple's Pages, etc...

Your programs should be formatted in a way that's readable. In other words, indent appropriately, use informative names for variables, etc… Points will be deducted if your indenting style makes it difficult to follow your code. If you are uncertain about what is a readable style, look at the examples from class as a starting point for a reasonable coding style. .

Your programs should compile and run without errors. Please do not submit programs that do not compile! It's better to submit partial implementation that compiles as oppose to implementations that do not compile.

At the top of each file you submit, you should put your name and your email as comments.

This homework set should be submitted via Sakai using the Assignment Menu option (on the left pane). You should only submit the requested ".java" files. Do not submit ".class" files or Eclipse-specific project files. Finally, please do not submit files using Sakai's Dropbox Menu.

Turn in the following file(s):

```
Season.java
FizzBuzz.java
Days.java
RandomCars.java
```

Directions

In this problem set, you will practice utilizing if statements along with methods and for loops.

Problem 1

In a class called `Season`, write a method called getSeason that takes as parameters two integers representing a month and day. This method then returns a String indicating the season for that month and day. Use the following dates for the seasons:

- 12/16 – 3/15 → "winter"
- 3/16 – 6/15 → "spring"
- 6/16 – 9/15 → "summer"
- 9/16 – 12/15 → "fall"

For example, `getSeason(2, 28)` should return "winter" and `getSeason(6, 16)` should return "summer". Write a main method to test your method with several test cases and print the results.

Problem 2

In a class named FizzBuzz, implement a program that plays a variation of the Fizz Buzz game. In this game, your program will print out numbers sequentially starting from 1 and ending at 30 except for the following 3 cases. For some given integers n and m:

- whenever the number to be printed is a multiple of n, print "Fizz!" instead,
- whenever the number to be printed is a multiple of m, print "Buzz!" instead,
- whenever the number is a multiple of both n and m, print "Have a Banana!" instead.

Sample output for n = 3 and m = 4 is as follows:

```
1
2
Fizz!
Buzz!
5
Fizz!
7
Buzz!
Fizz!
10
11
Have a banana!
13
. . . (the rest is omitted)
```

Your program should prompt the user for n and m where n and m should be different numbers between 2 and 9 (inclusive). The output of your program should match the example above when n = 3 and m = 4.

Problem 3

In a class named `Days`, write the following static methods:

- `int daysInMonth` – this method takes a single int parameter that is the numeric month and returns the number of days in the given month. A parameter of 1 would return 31 since there are 31 days in January and a parameter of 2 would return 28 since there are 28 days in February. (You may assume that it's not a leap year.) Your implementation should use logical ops (`&&` or `||`) and not use twelve `if / else if / else` blocks.
- `int dayOfYear` – this method takes two int parameters: month and day and returns the sequential day in the year. For example (1,1) is Jan. 1 and would return 1 since it's the first day of the year while (3,2) is Mar. 2 and would return 61 since it's the 61st day of the year. You should call the `daysInMonth` method from above in your method implementation.

Test your methods by calling them with several test cases in a main method and printing out informative messages indicating the results of the methods.

Problem 4

In a class called RandomCars write a program that draws random cars on a DrawingPanel window. You should write a static void method called drawCar that takes as parameters: a Graphics object g, and the x, y pixel coordinates where the car should be drawn. The x,y coordinates represent the upper left corner of the car. In the lecture notes on graphics we saw a class named Car. In that class the main method draws a car with a body, two wheels, and a window. You may start with that code and encapsulate it into the drawCar method. You may add extra features to the car if you wish.

To test your drawCar method, write a main method that creates a DrawingPanel window of size 400, 400 and then draws cars at random x and y coordinates in the window. After each call to drawCar, you should call the DrawingPanel's `sleep` method with a parameter of 200-500 msec so you can see the cars being drawn at a reasonable speed. After every set of 10 cars has been drawn, you should clear the screen using the DrawingPanel's `clear` method. Repeat this procedure 10 times for a total of 100 cars drawn. Your test main should only use a single for loop. (Hint: keep a count of the total number of cars drawn and use an if statement to decide when to clear the window.)