**CSCI 203: Introduction to Computer Science I**
**Bucknell University**
**Computer Science Department**
http://www.eg.bucknell.edu/~csci203

# Lab 9: More Decisions

November 2–3, 2009

# Table of Contents

## 1. Objectives

After completing the lab you should

1. Be able to construct more complicated Boolean tests
2. Be able to use Boolean operators.

**References:** The following references will be used throughout the lab.

- *CSCI 203 course website*
- *Java API website*
- Chapter 5 of Big Java

## 2. Preparation

Prepare for the lab by completing the following steps.

1. Start Eclipse.
2. Create a new Java project named `lab09-xyz01`, where `xyz01` is your login name.
3. Create the `readme.txt` file for this lab, insert the banner file with proper updates.

## 3. Introduction

In this lab we will explore the use of `if` statements and Boolean operators in Java. In each of the exercises you will be changing just *one* method. Do *not* change `main`!

## 4. Exercise 1

Import the file `Words.java` from

`~csci203/2009-fall/student/labs/lab09`

Complete the `getMiddle` method that gets the middle character from a word if the word length is odd, or the middle character pair if it is even. For example, `getMiddle("Java")` returns `"av"`. If the word is empty, return the empty string. If it is `null`, return `null`. The tests in `main` should pass when your method is complete. Place a copy of the program output in your `readme.txt` file.

## 5. Exercise 2

Import the file `MultipleChoiceTest.java` from

`~csci203/2009-fall/student/labs/lab09`

This class is used to determine if an answer on a multiple choice test is valid. A choice is valid if it is a letter of the alphabet between A and the maximum allowable choice. You specify the maximum choice value when you construct an instance of the class. For example, the following line constructs a multiple choice test with valid choices between A and E inclusive.

```
MultipleChoiceTest test = new MultipleChoiceTest("E");
```

The class `MultipleChoiceTest` contains a predicate method `isValidChoice` that determines if a choice is valid. A choice is valid if it is a *single character* between A and the maximum choice (inclusive), *ignoring case*. If a choice is `null`, it is not valid.

Your job is to complete the method `isValidChoice`. When the method is complete, all of the tests in `main` will pass. Place a copy of the program output in your `readme.txt` file.

## 6. Exercise 3

Import the file `Year.java` from

`~csci203/2009-fall/student/labs/lab09`

This class is used to determine whether a year is a leap year. A year with 366 days is called a *leap year*. A year is a leap year if it is divisible by 4 (for example, 1980). However, since the introduction of the Gregorian calendar in 1582, a year is not a leap year if it is divisible by 100 (for example, 1900); However, it is a leap year if it is divisible by 400 (for example, 2000). Note that the class `Year` defines a constant `GREGORIAN_START` that you may use.

Complete the predicate method `isLeapYear` so that it correctly determines whether a year is a leap year. All of the tests in `main` should pass when you are finished. Place a copy of the program output in your `readme.txt` file.

## 7. What to Submit

Your `readme.txt` file should include the output from exercises 1 through 3. Make sure your `readme.txt` file is in your lab directory, and drag your lab directory to the CSCI 203 lab drop box: `CSCI203 Lab drop_box`