CS172 LAB4 – Dates
Due Friday, February 20 (1:15pm)

In this lab, you will practice using the control structures that help to make decisions: if statements, boolean variables, and switch statements.

General Instructions:

1. Prepare your lab report in an essay format, answering every numbered question in the lab instructions below.

2. The report should have a cover page containing your name, the lab number, the date of the lab and the date that the report is submitted.

3. Evaluation will depend on the clarity of your writing, the correctness of your answers, and the adherence to the lab instructions.

Please submit your lab report by email to walkerel@hiram.edu by the deadline. Unless you have difficulty with the email submission, do not submit a hard copy in class. In addition to the written report (.doc file or equivalent), you should submit zip archives of all BlueJ projects that you create in this lab.

Lab Instructions:

For each step of this lab, write a short paragraph describing the implementation process for that object or method, any problems that you ran into, and how you solved them.

1. In a new project, create a class called Date(). A date consists of a year, month, and day. All three of these should be integers.

2. Make a constructor for your class. This constructor should have the year, month and day specified as parameters. Your constructor should require that the month be in the range 1 (January) to 12 (December) and that the day should be in the range 1 to 31. (We will change this later to reflect the actual number of days in a month.) If the month or day is bad, set the day to 0.

3. Make a method called toString for your Date class. It should take no parameters and should return a string like “February 11, 2004” for the current date. If the day is 0, it should return the string “Invalid Date”.
4. Make a predicate method called isLeapYear for your Date class. Every year that is evenly divisible by 4 is a leap year (like 2004). One exception is that a year is not a leap year if it is divisible by 100 (like 1900). But there is an exception to the exception that if a year is divisible by 400, then it is a leap year. So 1600, 2000 and 2400 are leap years, but 1700, 1800 and 1900 are not leap years. Your method should return true if the year is a leap year.

5. Make a method called daysInMonth that returns the number of days in the month. The correct number of days is: 31 for January, 28 for February, unless it is a leap year, 31 for March, 30 for April, 31 for May, 30 for June, 31 for July, 31 for August, 30 for September, 31 for October, 30 for November, and 31 for December. (This method will probably use the isLeapYear method). Use a switch statement to implement this method. (See Section 5.2).

6. Revise your constructor to use the daysInMonth method to determine whether the date is valid. Also write a function called changeDate that takes a day, month and year as parameters and if the date is valid, changes the date and returns true. If the date is not valid, it should make no changes and return false.

7. Write a method to find the difference in days between this date and another date. If the dates are equal, then the result is 0. If the current date (this) is before the parameter date (that), then the answer should be positive. Otherwise, the answer should be negative. Your first try at this function should estimate the number of days by adding 365 days for each year of difference and 30 days for each month of difference. For example, the difference between 2/19/2004 and 4/19/2005 would be estimated as 365 (to 2/19/2005) + 60 (to 4/19/2005) + 1 to 4/20/2005 = 426. Call this function estimatedDaysUntil.

8. (Extra credit: 5 points) Write a method that determines the exact number of days until another date. Your method needs to take into account leap years (not every year has 366 days) and the different number of days in each month, including the 28 or 29 in February as appropriate. This function should be called actualDaysUntil.

9. Write a testDate class with a main function that shows off all of the abilities of your Date class. At the very least, it should let you input 2 dates (as 3 integers each), print them out in a string format, tell whether the years are leap years, and tell the estimated or actual number of days from the first date until the second date.