CS150 Lab 4

Leap Years

Date Assigned: Tuesday, September 20, 2005

Date Due: Thursday, September 22, 2005

Points: 15

Objectives

For this lab, you are to write your own program that solves the following problem. You will need to use selection structures and logical operators.

Problem statement

For an inputted year, your program must print out whether it is a leap year or not. Here are the rules:

A year is a leap year if it is divisible by 4. The only exception to this is if it is a century year; then it is a leap year only if it is divisible by 400. In case you're wondering, these are the rules for the Gregorian calendar, which began to be adopted in 1582 when they realized that having a leap year every 4 years resulted in the days of the year being very off after a several centuries. (The actual length of a year is 365.24219 days, not an even 365.25 days). It's amazing what information you can find on the Internet if you look!

Sample output

Leap Years

```
Enter a year:  1900
1900 is not a leap year
```

Steps for software development

1. First, we need to understand the program requirements. What needs to be calculated? Is there any additional information that we need?

2. Next, we need a program analysis. Answering the following questions will help guide you through the process.

(a) What is the input to your program? What units will it be in?

(b) What is the output to your program? What units will it be in?

(c) Is there any data that will be internal to your program?

(d) What are the calculations needed for your program?

3. What is the algorithm to solve this problem? Here you should describe in English the steps for solving the program. This is the place where you decide the specifics of your program. For example, if you need to use the `if` selection structure or logical statements.

4. Create a new project in Visual Studio .NET. You should name your project "04LeapPUNetId", where PUNetId is your own id. I would name my project "04Leapkhoj0332". While working on a project, it should be located on the current computer you are working on (i.e. the desktop). Once you have completed developing, you should copy the project folder onto Turing.

5. Write the code that will solve the problem. Make sure that you add comments to the code as you type and that your code follows the coding standards

6. How can you verify that your program works correctly? What numbers would you use to test the program. Make sure to test for every possible case.

What to turn in

When you have completed writing the program and you have verified that it works correctly, you will need to show it to the instructor or the TA.

Once you have done this you will submit the project for grading. You submit your program by placing a copy of the project folder in the "CS150-01 Lab" folder on Turing. Make sure that you also place a copy of the project folder in your own folder on Turing.