

CS360 Windows Programming

Assignment 1

Date assigned: Thursday, 15 September 2005

Date due: Tuesday, 27 September 2005

Points: 25 points (15 pts for part 1, 10 pts for part 2)

Your first assignment will get you ready to do serious work with C# and .NET. There are two parts:

1. Write a simple C# console application that calculates mortgage payments.
2. Use Visual Studio to write a GUI version of the mortgage calculator from Part 1.

Before getting into the details, I want to outline some of the philosophy behind the assignments in this course. What you will be doing is learning new technology, which is a very common occurrence in today's computing environment.

Getting up to speed quickly is a skill that is well worth developing, and to do that you need to be an active participant. You need to be energetic and inquisitive, and you shouldn't expect everything to be spelled out in the ultimate detail. I'm here to help, but your basic response to a situation you don't understand should be "how can I figure that out?" rather than "who can give me the answer to that?"

Part 1: A simple mortgage calculator – console version

Goal: Write a simple C# application that calculates mortgage payments. This is going to be a text based console application, and a run of your program might look like:

```
Welcome to the Mortgage Calculator.
You'll be working with four quantities:
    mortgage amount
    time period of mortgage
    rate of interest
    monthly payment
Given 3 quantities, the Mortgage Calculator will compute the fourth value

What would you like to compute? <Enter 'q' to quit>
<M>ortgage Amount, <T>ime Period, <R>ate or Monthly <P>ayment?
p
Enter mortgage amount: 100,000
Enter time period (in years): 30
Enter interest rate (in percent): 5

The monthly payment amount is: $536.82

What would you like to compute? <Enter 'q' to quit>
<M>ortgage Amount, <T>ime Period, <R>ate or Monthly <P>ayment?
r
Enter mortgage amount: 100,000
Enter time period (in years): 30
Enter monthly payment: 500

The rate of interest is: 4.39%

What would you like to compute? <Enter 'q' to quit>
<M>ortgage Amount, <T>ime Period, <R>ate or Monthly <P>ayment?
t
Enter mortgage amount: 100,000
Enter interest rate (in percent): 5
Enter monthly payment: 500

The time period (years) for repayment is: 35.91

What would you like to compute? <Enter 'q' to quit>
<M>ortgage Amount, <T>ime Period, <R>ate or Monthly <P>ayment?
q
```

In this run, the user first computed the payments on a \$100,000 mortgage for 30 years at the interest rates of 5%, then determined what rate would reduce the payment to \$500; and then determined how many years the mortgage would have to run to have \$500 payments at a rate of 5%.

You will write your code in C#. For this part of the assignment, any text editor will do, even Notepad. You will compile your code using the command line compiler, and then run it from the DOS prompt.

It's important that you realize that those are the only tools you need. Once you are into fancier things, like visual design of a GUI, it's easy to get the idea that Visual Studio is .NET. That is certainly not the case. You could go through the whole quarter using Notepad and the command line compiler if you wanted to.

Having said all that, we would actually suggest that you go ahead and use Visual Studio for this part, but just as a program editor. Open VS and create a new text file. Once you type a line or two, save the file with the extension ".cs". Now VS will treat it as a C# file

and give you color-coding and other niceties that you wouldn't get from Notepad. When you are ready and have saved your file, you will compile it in a DOS command window (discussed below).

I don't want you to spend your time learning how to do financial calculations, so I am providing you with a precompiled library: `MortgageCalculatorFunctions.dll`¹. This library defines a namespace called `MortgageCalculatorFunctions` that defines a single class called `CIFunctions`. (CI stands for Compound Interest.) That class has four static methods that compute mortgage amount, number of payment periods, rate, and payment.

Documentation for the library is provided to you as HTML. When you have unzipped the distribution file (available in the CS360 folder on Turing), look in the folder `CodeCommentReport` and click on `Solution_MortgageCalculator.HTM`. You then have to drill down a little, but you will find documentation on the class as a whole and on each method.

The compiler is called `csc`, and it is probably stored in a directory something like this:

```
C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322
```

You'll want to be sure that that directory is in your Path. I'll talk about how to do that and how to use the compiler in class. Basically, you will invoke the compiler with a command like this:

```
csc /r:MortgageCalculatorFunctions.dll MyCommandLineCalculator.cs
```

That assumes you have copied the dll into the same folder as your code. The result will be a file called `MyCommandLineCalculator.exe` that you can run.

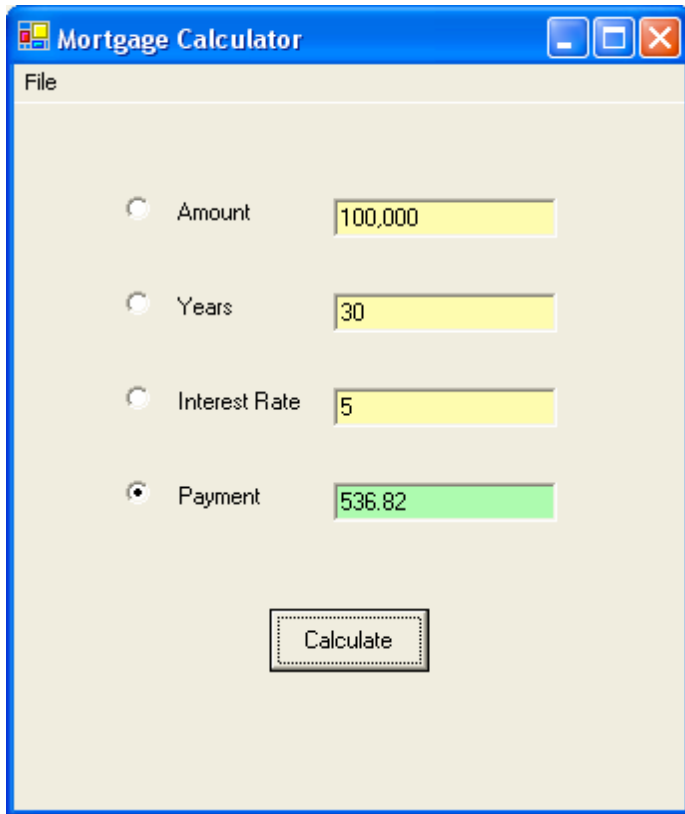
Your program should gracefully handle cases where the user provides a bad input (e.g., "xyz" for the mortgage amount) or where the financial code throws an exception.

Part 2: A simple mortgage calculator – GUI version

Your next step is to open Visual Studio and start a new project (from the File menu). This should be a C# project, and you should use the template called Windows Application. Choose your own name and location for the project.

Your goal is to write a GUI-based application that calculates mortgage payments. This will give you experience in doing visual design, using Windows controls, and writing a little more C#. Your final product should look something like this:

¹ Thanks to Stanford University for this dll



Here are a few things to note:

- You will have to add a reference to `MortgageCalculatorFunctions.dll`. Do this by rightclicking References in your project in the Solution Explorer panel and choosing Add Reference. Click on the Projects tab, then navigate to the dll. Do this early-on, and add the line `using MortgageCalculatorFunctions;` to the beginning of your code. At that point IntelliSense will be able to help you with the `CIFunctions` class.
- Fields are aligned and have labels.
- The radio buttons are used to indicate which value is being calculated.
- The calculated field should have a different background color from the other fields, and this should be kept up to date as different radio buttons are pressed. You can set the initial colors in design mode, and you should decide which field to calculate by default
- When a radio button is pressed, the corresponding field should be blanked, indicating that the Calculate button has not been pressed for this selection.
- The application window has a descriptive title.
- There is a `File` menu. The only item that needs to appear there is `Exit`. The code behind the menu item is simply a call to `Application.Exit()`.

Submission Instructions

- Submit a folder with both versions of your program to the CS360 drop folder by 9.25am on the day the assignment is due.
- Print out a hardcopy of all of your code and hand it in on the due date.