W. P. CAREY SCHOOL of BUSINESS
ARIZONA STATE UNIVERSITY

CIS 345 – Business Information Systems Development II – Fall 2014

Assignment 1: Loops

Due on Blackboard: Friday, September 5 by 10 PM

Blackboard submission links will disappear after the due date/time and you will not be able to submit assignments late. No late submissions will be accepted (either throughout Blackboard or via e-mail). Do not wait till the last 5 minutes to submit.

Have a submission turned in at least a few hours before it is due even if you are not done. Resubmit again later if you needed to. Blackboard will allow resubmissions. Previous attempts will be cleared and not graded (or even looked at). Only the final submission will be graded.

Objective: This assignment is meant to give you practice in creating logical structures using programming loops.

Implement the following as a Visual C# Console application. Name your project Assignment1 and your class Loops.cs. Put your name, class, and class time within the class file as a comment.

Within the *Main* method of your application, write the logic for the following parts. Precede each part with a line identifying which part is being displayed, e.g., "CIS 345 Assignment 1, Part 1 – While Loop."

1) Using a *while* loop, list all odd numbers from 1 to 20 on the Console window. Start from 1 and jump directly to the next odd number by incrementing your counter variable appropriately. *This loop must not iterate 20 times. (2 points)*

2) Using a *for* loop, list all even numbers from 1 to 20. In this problem, your counter variable must move from 1 to 20 and it must assume all values from 1 to 20 i.e. you cannot make the counter variable skip from 2 to 4. *This loop must iterate 20 times.*

   The remainder operator (%) gives you the remainder. Use the remainder operator and an "*if"* statement to determine if the number is even. *(2 points)*

3) Do Exercise 5.37 from *Visual C# 2012: How to Program*. You will have to write a program that computes the factorial of a number e.g. 4! = 4 * 3 * 2 * 1 = 24.

   Use either a *while* or a *for* loop. You will need to ask the user for a number and then utilize that as the starting value of your loop counter. Count down and keep multiplying while your counter is above zero. *(2 points)*

4) Create the triangles in Exercise 6.15 from *Visual C# 2012: How to Program*. Do only parts (b) and (d). *(3 points each).*

5) Create a multiplication table that shows the numbers 5 through 10 multiplied by each number in the set {10, 20, 30, 40, 50}. Show the multiplication equation as well as the product:

   ```
   5  *  10    =    50
   5  *  20    =   100
   5  *  30    =   150
   5  *  40    =   200
   5  *  50    =   250
   …and so on.
   ```

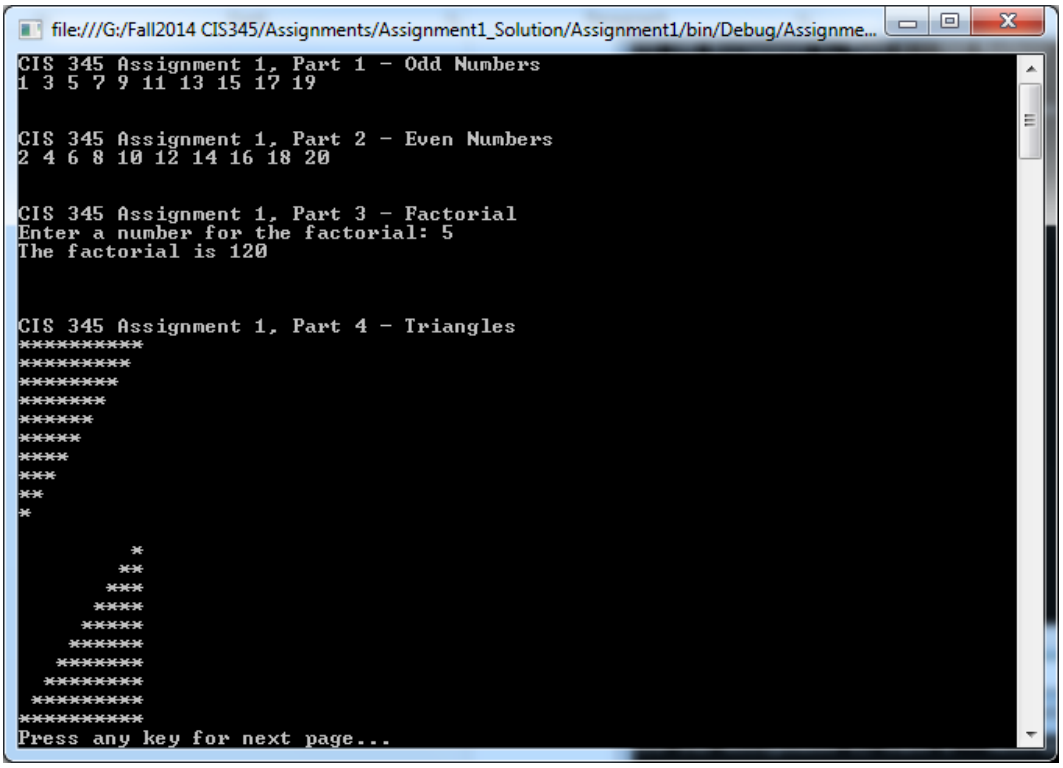
   Produce a clean tabular format by using place-holders as shown in the sample output. All of your values should be inserted into your *Write* statement using place-holders. Refer to Fig 6.6 on page 197 in *Visual C# 2012: How to Program* for an example of using place-holders. An explanation on using place-holders follows on page 198. You might need to use a mix of right and left justification. *(3 points)*

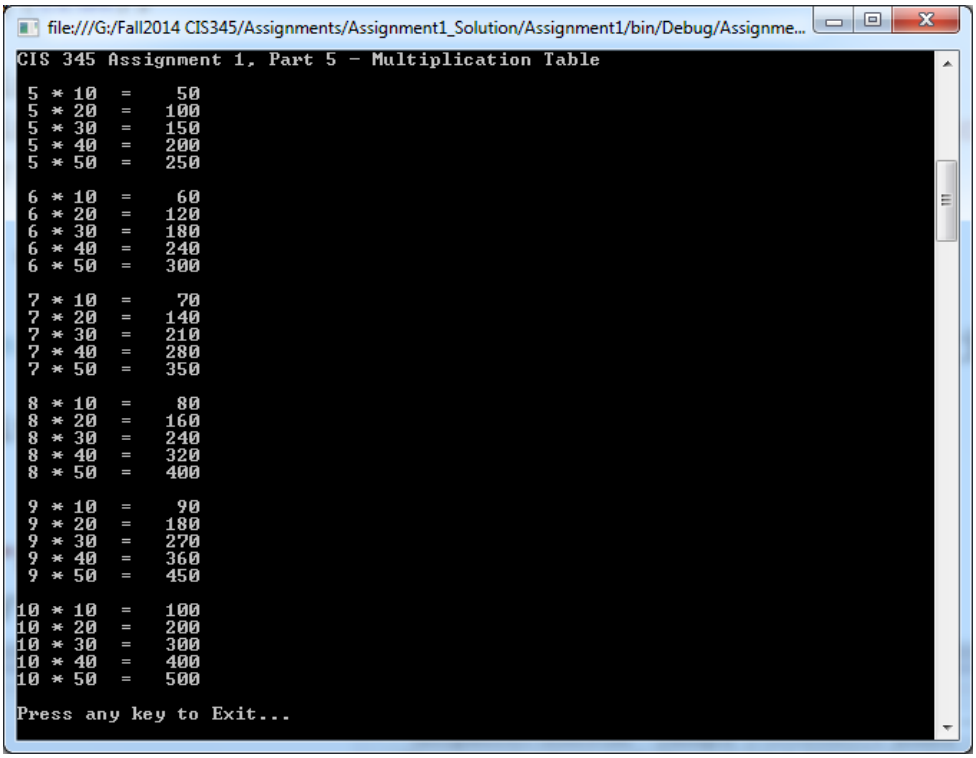
You do NOT need to implement different methods for this assignment. However, if you choose to do so, you may write each part in a different method.

Note:

1. CIS 345 assignments are graded on source code AND output, with the emphasis on the code.
2. There is no need to create release builds (since the emphasis is on the code).
3. Slight deviations in formatting are fine (e.g. if tabular output is required, then you should have tabular output but it doesn't matter if your columns are wider/narrower). Don't spend excessive time in trying to get every character to match. *There is a slight emphasis on matching formatting in Assignment 1 since creating patterns is the objective of the exercise. Otherwise, your output can vary in formatting in future assignments.*
4. Assume perfect input by user – no validation is necessary. Also, do not worry about the program crashing due to large numbers (e.g. when doing the Factorial part.) *We will handle these issues in the second half of the semester.*

Sample Output

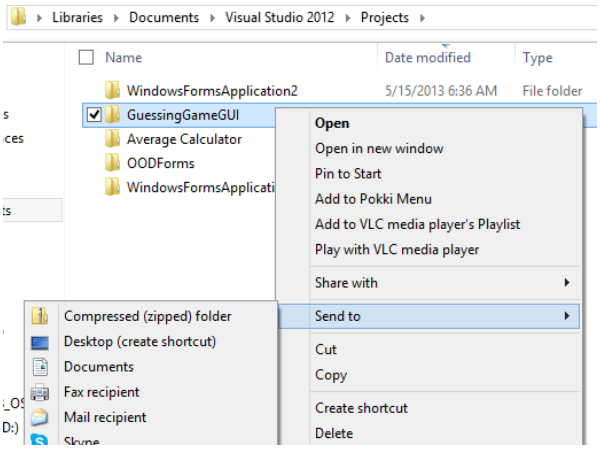




Submission Instructions

Submission should be made using a zip file that contains all of the Visual Studio C# project files. You will need to ***zip the entire project folder*** along with the .sln and .suo files. The folder will automatically contain the class source files as well as the executable file that is generated in \Assignment1\bin\Debug folder. Upload file to the Blackboard assignment drop box.

Zip the entire top-level folder by right-clicking the folder and selecting Send to | Compressed (zipped) folder.



Using built-in windows zip tools: http://windows.microsoft.com/en-US/windows-vista/Compress-and-uncompress-files-zip-files/

Make sure you check the following. Your grade is dependent on all these criteria being met.

- You have included your name and class as a comment within your class file.
    - e.g. " // Assignment 1, Jane C. Smith, CIS 345, Tuesday 9:00 AM"
- Class file is called Loops.cs (rename from Program.cs).
- Your Visual Studio project is called Assignment1.
- Zip filename is: Assignment1.zip

Verify your zip file before you submit

- Check for actual class files being present in the folder before you zip it.
- Check your zip file size after zipping – if it is 1K, it likely contains only a shortcut.
- Uncompress your zip file before submitting and verify that files are present.
- Download your zip file after submitting, uncompress, and again verify that your files are present.

This takes an extra couple of minutes. Please do it if your grade is important to you. If you do this, you will not end up submitting a bad file. If you submit an empty file, or one containing only a shortcut, or a bad zip file, you will receive a score of zero and your only recourse will be to do the makeup assignment at the end of the semester.