

CSCI 1016 Assignment 1

Q1: Textbook Page 41, Chapter 1, Programming Challenges **1, Carpet Size** [30 points]

Q2: Textbook Page 41, Chapter 1, Programming Challenges **4, Account Balance** [30 points]

Note: For Q1 and Q2, you do NOT need to do anything in Visual Studio. Instead, put your answers in a Microsoft Word document **lastname_firstname_assignment01.doc**
... or a pdf file **lastname_firstname_assignment01.pdf**.

The challenges require a small drawing or sketch. Create this sketch anyway you want. Some ways to draw it might be:

- Draw it on paper, take a photo of it or scan it, insert it to Word and/or convert it to pdf
- Use dashes and underlines in Word
- MS Paint or other graphics software

The sketches do not need to be works of art. They should be legible and clearly indicate the point you are trying to make.

Q3: Textbook Page 105, Chapter 2, Programming Challenges **2, Name and Address** [40 points]

Note: For Q3, name your project <my name> Name and Address. It is ok to put spaces in the project name. For example, my project would be named: Leong Lee Name and Address.

Follow all the textbook requirements on Challenge 2, 2a to 2d.

Update to 2d: The form's title bar should read <my name> Name and Address. For example, my form's title bar should read Leong Lee Name and Address.

Important: If you do not put <my name> to the above mentioned fields (project name and form title bar), **you will get 0 point for Q3.**

Submit your zipped solution **lastname_firstname_assignment01.zip** to the D2L Dropbox. Make sure to zip up the entire solution directory.

Submission instructions:

Please submit electronic copy (**two files**) to D2L digital dropbox.

Name the files ...

1. **lastname_firstname_assignment01.doc** or **lastname_firstname_assignment01.pdf** (for Q1 & Q2)
2. **lastname_firstname_assignment01.zip** (for Q3)

Grading guidelines (programming questions):

Your programs will be judged on several criteria, which are shown below.

- Correctness (50%): Does the program compile correctly? Does the program do what it's supposed to do?
- Design (20%): Are operations broken down in a reasonable way (e.g. classes and methods)?
- Style (10%): Is the program indented properly? Do variables have meaningful names?
- Robustness (10%): Does the program handle erroneous or unexpected input gracefully?
- Documentation (10%): Do all program files begin with a comment that identifies the author, the contents, and the compiler used for that particular file? Are all the classes, methods and data fields clearly documented? Are unclear parts of code documented? (Some items mentioned may not apply to some languages)

A program that does not compile will get at most **50% of the possible points**.