**1090C Computer Programming I  
Lab 1 – Flow Charts and Pseudo Code  
SPRING 2020  
10 pts / 2 extra pts**

*Labs are never about getting done… They are about learning.   
  
I often will include a short mini-lecture at the start of the lab directions. This is designed to focus your attention on the learning goals for the lab and often provides you with specific technical details you will practice directly in the lab session.*

**Mini-lecture:**  
We use pseudo code and flow charts to capture the logic of a program prior to implementing it with code. In general practice, flow charts are too cumbersome for large programs/systems and so they are mainly used for isolated sub-routines, and high-level architectural patterns, etc.   
  
We use pseudo code all the time. The pseudo code outline we write initially becomes the documentation for the completed program. To code the program, we go through and write the program code that corresponds to the pseudo code outline. The outline then becomes the program comments that document the code.  
  
Students rarely believe me initially when I mention that the important part of this process is the generation of the pseudo code outline. Generating the code that corresponds to the outline is trivial and simply a matter of practice with the language syntax. If the pseudo code outline is incorrect then the program code will not correctly solve the task. Instead of trying to “write the program out of your head” as it were, take the time to create a solid and logically clear pseudo code outline before you begin coding.

**Common flowchart symbols:**

* + Lines (Arrows) show the flow
  + Input symbols (parallelograms)
  + Processing symbols (rectangles)
  + Output symbols (parallelograms)
  + Terminal symbols (lozenges – flattened ovals)
  + Decision symbols (diamonds)

**output “What is your favorite Number?”**

**doubleFun = favNumber \* 2**

**output “Double your fun is: ” + doubleFun**

**input favNumber**

**Pseudo Code:**

Pseudo code is informal and fairly close to natural language. Rather than complete sentences, we tend to use very brief statements which will ultimately translate more directly to program code. Indentation is used to show groups of related statements, pay attention to it.

Look carefully at the example here and be sure to understand how both the pseudo code and the flowchart express the same logic (i.e. the same program).

**Prompts:** In every case, you have to **prompt** the user so they know what you need for them to input. Thus, before every input, we output the prompt msg so the user knows what we want to get from them.  
  
**Always include the prompt in your pseudo code and flowcharts when you do input.**

**Output should always be in the form of a complete sentence, not just a raw calculated value. i.e.** *The total calculated building costs is: $560.00.*

**Example:**

**start  
 output “What is your favorite number?”**  
 **input favNumber  
 doubleFun = favNumber \* 2  
 output “Double your fun is “ + doubleFun  
stop**

**Lab:**

1. Insert your work for the completed lab here within this MS Word document and submit it as directed at the end of the document.
2. For each of the following tasks, provide **both** the flow chart and pseudo code. You may use the MS Word symbols to create the flow charts or any other tool that lets you create and embed them in this document. I’ll likely work through the first task with you during the session. Just insert your response right here in this document. Before doing the flow chart or pseudo code, list the input(s) and output(s) required for each program. (Please note that these are all simple programs with no branching so the flow charts will be a vertical connection of symbols like the example here.)  
   1. **Task 1 (2 pts): a program where the user enters the price of a purchase and the program computes and outputs a 5% sales tax. Don’t forget the prompt!**

**start  
 output “What is the purchase price?”**  
 **input purchasePrice  
 salesTax = purchasePrice \* .05** **output “5% sales tax on your purchase price is “ + salesTax  
stop**

**output “What is the purchase price?”**

**input purchasePrice**

**salesTax = purchasePrice \* .05**

**output “5% sales tax on your purchase price is “ + salesTax**

* 1. **Task 2 (2 pts): A program for calculating the area in square feet. (User will input the height and the width in feet.)**

**start  
 output “What is the height in feet?”**  
 **input height**

**output “What is the width in feet?”**

**input width  
 area = height \* width   
 output “The area is “ + area “ feet squared”   
stop**

**output “What is the height in feet?”**

**input height**

**output “What is the width in feet?”**

**input width**

**area = height \* width**

**output “The area is “ + area “ feet squared”**

* 1. **Task 3 (3 pts): A program that asks the user to enter home maintenance costs for each of the four seasons and returns the total yearly maintenance costs. (Separate prompt and input for each input value!)**

**start  
 output “What is your home maintenance cost for winter?”**  
 **input winterCost**

**output “What is your home maintenance cost for summer?”**

**input summerCost**

**output “What is your home maintenance cost for spring?”**

**input springCost**

**output “What is your home maintenance cost for fall?”**

**input fallCost  
 yearlyCost = winterCost + summerCost + springCost + fallCost  
 output “Your yearly home maintenance cost is “ + yearlyCost  
stop**

**output “What is your home maintenance cost for winter?”**

**input winterCost**

**output “What is your home maintenance cost for summer?”**

**input summerCost**

**output “What is your home maintenance cost for spring?”**

**input springCost**

**output “What is your home maintenance cost for fall?”**

**input fallCost**

**yearlyCost = winterCost + summerCost + springCost + fallCost**

**output “Your yearly home maintenance cost is “ + yearlyCost**

* 1. **Task 4 (3 pts): A program that calculates the difference between two numbers. i.e. subtracts one from the other. Thus this might be a negative value.**

**start  
 output “Input any number”**  
 **input firstNumber**

**output “Input a second number”**

**input secondNumber**

**difference = firstNumber - secondNumber  
 output “The difference between these numbers is “ + difference  
stop**

**output “Input any number”**

**input firstNumber**

**output “Input a second number”**

**input secondNumber**

**difference = firstNumber - secondNumber**

**output “The difference between these numbers is ” + difference**

* 1. **Task 5 (2 pts Extra Credit): A program where the user enters a credit card balance and the program calculates the interest at a rate of 17% monthly. Assume the user does not make any payment. Display the interest due after one month and again after two months.**

**start  
 output “What is the credit card balance?”**  
 **input cardBalance  
 interest = .17**

**oneMonth = cardBalance \* interest**

**twoMonth = (cardBalance + oneMonth) \* interest  
 output** **“Interest due after one month is “ + oneMonth “, and “ + twoMonth “ after two months”   
stop**

**output “What is the credit card balance?”**

**input cardBalance**

**interest = .17**

**oneMonth = cardBalance \* interest**

**twoMonth = (cardBalance + oneMonth) \* interest**

**output “Interest due after one month is “ + oneMonth “, and “ + twoMonth “ after two months”**

1. Submitting your work: carefully check your work.   
     
   **You need to follow directions and develop professional work habits. I’m very specific about how I want the work submitted especially with the naming and the format of files so they can be tracked efficiently. If you can’t or won’t follow directions, you work will not be graded and you receive no credit.**

Rename your copy of this word file as **Lastname\_Firstname­\_Lab01.docx** using your name. Submit this file using the Canvas assignment mechanism. Submit the exact same file a second time using the additional Canvas link for the extra credit option.