

## Assignment 2 – Control Statements/User-Defined Functions

Deadline: Friday Oct. 12 at 23:55  
Type: Individual Assignment  
Weight: 5%

### Submission instructions:

- Create a cpp file for each question
- Compress the files using zip or other tools
- Submit the zip file on Moodle

### Notes:

- Please do not submit exe files
- All submissions must be done through Moodle

### Marking Scheme:

- Program correctness (80%)
  - Program clarity (output format, comments, completeness, readability) (20%)
- 

### Exercises:

1. Write a C++ program to assist a university administrator in determining some statistics about students. For each student, the administrator enters the student number (5 digits), gender (1 for male and 2 for female), age, program of study. The administrator wants to determine the total number of students, the number of male students, the number of female students, etc. The administrator does not know in advance the number of students. Using your program, she or he should be able to indicate that all student information has been entered by entering the sentinel value -1 as the student number. The output of your program may look like this:

```
Enter student number: 12345
Gender (1=male, 2= female): 1
Age: 18
Program of study: COEN
```

```
Enter student number: 33345
Gender (1=male, 2= female): 2
Age: 20
Program of study: COEN
```

```
Enter student number: 54345
Gender (1=male, 2= female): 2
Age: 18
Program of study: ELEC
```

```
Enter student number: -1
```

```
Statistics:
Total number of students is: 3
Number of male students is: 1
Number of female students is: 2
Average age of students is: 18
Number of COEN students: 2
Number of ELEC students is: 1
```

2. Sorting data is an important task that many data storage systems use. This problem will be based on the bubble sort. The bubble sort will compare adjacent entries and swap them if the entries are not in the correct order. This is repeated until all the data is in the correct order. Write a C++ program that will compute that will perform the following operations. It will require that the person using the code enter six numbers sequentially (one number at a time). After all the numbers have been entered, the code will use a bubble sort method to ensure that the numbers are sorted in ascending order (smallest to largest).

This program will read six numbers typed in by you

And then sort and print out the sorted numbers

Please type in the first number

6

Please type in the second number

5

Please type in the third number

4

Please type in the fourth number

3

Please type in the fifth number

2

Please type in the sixth number

1

1 2 3 4 5 6

This program will read six numbers typed in by you

And then sort and print out the sorted numbers

Please type in the first number

2

Please type in the second number

3

Please type in the third number

4

Please type in the fourth number

2

Please type in the fifth number

3

Please type in the sixth number

4

2 2 3 3 4 4

Sample Output

3. Write a C++ program that will compute the roots of a user defined function (which will be generated by the student). The user defined function is:  $f(x) = (x - e^{-x})$ . You must edit the code to include the user defined function and insure that the code compiles and has no runtime errors.

```
// COEN 243 Base program for Ass#2. Q3
// Sept. 22 2018.
//
// This program will use the secant method to
// compute the root for a function specified
// in the main code block. Update this later
// when functions are covered in lecture

#include <iostream>
#include<cmath>
//
using namespace std;

int main ()
{
//
// Define the values
//
// ST_A: User provided starting value A
// ST_B: User provided starting value B
//
// These values are required to begin the iteration as
// difference approximation of derivative requires two points
//
// Min_Mag_F: User provided minimum magnitude of the function
// value. Effectively how close to root is acceptable
// within the numerical resolution of the system.
//
// Internal values
//
// CNTR: Used to keep track of iteration number
// X_New: Latest estimate of the root
// X_Now: Previous estimate of the root
// X_Pre: The oldest estimate of the root
// F_X_New: The function using the latest estimate of the root
// F_X_Now: The function using the previous estimate of the root
// F_X_Pre: The function using the oldest estimate of the root
// New_Mag_F: The magnitude of the function computed using
// latest estimate of the root
//
int CNTR;
float X_New, X_Now,X_Pre;
float ST_A, ST_B, Min_Mag_F;
float F_X_New, F_X_Now, F_X_Pre;
float New_Mag_F;
```

```

//
// Ask for the user to input the starting values and
// the termination conditions
//
// Function in this case will be user defined
//
cout << "This program will compute a root of a pre-set function " << endl;
cout<< " you will enter two starting values for search of root and " << endl;
cout << " and a ending condition of how small magnitude you will allow."<<endl;
cout <<endl;
cout << "Please type in the value of the first start point " << endl;
cin >> ST_A;
X_Pre=ST_A; // Load starting value into internal Var.
cout<<endl;
cout << "Please type in the value of the second start point " << endl;
cin >>ST_B;
X_Now=ST_B; // Load startin value into internal Var.
cout <<endl;
cout << "Please type in the smallest magnitude of the function " <<endl;
cin >> Min_Mag_F;
if (Min_Mag_F < 0) Min_Mag_F=-1*(Min_Mag_F); // Pos Magnitude
F_X_Pre=Ustr_Fun(ST_A); // Compute f(ST_A)
F_X_Now=Ustr_Fun(ST_B); // Compute f(ST_B)
cout<<"FA=" <<F_X_Pre<<" FB= " <<F_X_Now<<endl;
New_Mag_F=1000000; // Intialize ending conditions to dummy value
while (New_Mag_F >= Min_Mag_F)
{
    CNTR++;
    X_New=X_Now-(F_X_Now*(X_Pre-X_Now)/(F_X_Pre-F_X_Now)); //Estimate
    F_X_New=User_Fun(X_New); // Compute Function at root estm.
    cout << "Estimate of root #" <<CNTR<<" is " << X_New << endl;
    cout << "The function is: " <<F_X_New<<endl;
    if (F_X_New < 0) New_Mag_F=-1*(F_X_New);
    else New_Mag_F=F_X_New;
    X_Pre=X_Now; // Update the values
    X_Now=X_New; // Update the values
    F_X_Pre=User_Fun(X_Pre); //Update the functions
    F_X_Now=User_Fun(X_Now); //Update the function
    if (CNTR >= 100) break; // exit from stuck in loop
}

return 0;
}

```

Code that must be modified by the student