Labsheet 9 - Programs using Loops

You should be getting **3 programs done per lab session at a minimum**. If you don't you should work on them outside of lab time and have them done before the next lab session.

Remember to design your programs with pseudo-code and/or flow diagrams first. Name the programs appropriately.

It is absolutely prohibited to copy anyone else's design or code. You can ask for help with a particular problem from friends, colleagues, the lecturers in the lab etc but you must write your own design and code the fix to the problem yourself.

If you get help from someone or take design/code from the web or elsewhere, you have to comment in your code to state what help you got and from where.

Saving Your Projects

You can create a new project for each program or create them as separate files within the one project.
You should save the programs to a folder called **labsheet 9** within **Programming\Labsheets** folder on your **M drive.**

Every program should be properly ***commented*** including having a comment at the top stating your name, student ID number, date created, approx number of hours worked on, overall brief description of the program and any known bugs in it.

Q.1

a. Write the pseudocode algorithm for this function. Write a function called **displayNumbers()** which will display the numbers 1 to 15 on the screen using a loop. Your function should also **sum** these numbers and return the sum to the main() function and display the sum on the screen within main().

b. Next code the program.

c. Test your code.

Q.2

a. Write the pseudocode algorithm for this function. Write a function called **reverseDisplayNums()** which will display the numbers 30 to 20 on the screen using a loop. Your function should also **sum** and find the **average** of these numbers. It should display the sum and return the average to the main() function and display the average on the screen within main().

b. Next code the program.

c. Test your code.

Q.3

a. Write the pseudocode algorithm for this function. Write a function called **generateRandom()** which will randomly generate 10 numbers using a loop. Use the ***rand( )*** function to generate the random numbers in the range 1 to 40. Your function should display and **sum** the numbers.

   See your notes **Chapter 4** for information on random number generation.

b. Next code the program.

c. Test your code.

Q.4

a. Write the pseudocode algorithm for this program. Write a function called **generateRandomEven()** which will randomly generate 10 **EVEN** numbers using a loop. Use the ***rand( )*** function to generate the random numbers in the range 10 to 50. Your function should **sum** the even numbers and return the sum to the **main()** function. The main function should display the sum on the screen.

b. Next code the program.

c. Test your code.

Q.5

a. Write the pseudocode algorithm for this program. Write a program which will **ask the user** the number of numbers they want to input. This should be done in the **main()** function. An error message should be displayed in the main() if the user enters a number <= zero.

This number is passed to a function called **calculateNumbers()** which should then read that number of numbers from the user. The function **calculateNumbers()** should **sum** the numbers entered and display it. It should also **count** and **sum** the ***odd numbers*** entered and display the count and sum of the odd numbers entered to the screen.

There is an executable program of this question on your shared OneDrive called Q5 Program.exe. Download and run this program and you will see how it works.

b. Next write test data for this program.
c. Next code the program.
d. Test your code against the test data.

Q.6 Multiplication Table

a. Write the pseudocode algorithm for this program. Write a program which asks the user to enter a table number in the main( ).

Enter a table number :

Read the input and display an error message if the number entered is not between 1-12 (inclusive) and the user should be asked to re-enter.

The program should then call another function **displayTimesTable()** which displays the multiplication table for that number which is passed as an argument to the function.

For example if the number 4 is entered:
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12 etc.

There is an executable program of this question on your shared OneDrive called Q6 Program.exe. Download and run this program.

b. Next write test data for this program.
c. Next code the program.
d. Test your code against the test data.

Q.7

a. Write the pseudocode algorithm for this program. The program will **ask the user** the number of numbers they want to input. This should be done in the **main()** function. An error message should be displayed in the main() if the user enters a number <= zero and the user should be asked to re-enter.

This number is passed to a function called **calculationsNum()** which should then read that number of numbers from the user. This function should **sum** the numbers, **find the largest and smallest numbers** of the numbers entered and display this information on the screen. It should also find the **average** of the numbers entered and return the average number to the **main()** function. The main function should display the **average number** on the screen.

The numbers entered by the user must be in the range 5-15. An **error message** should be displayed if the user enters a number outside the range. The user should be asked to reenter the number.

There is an executable program of this question on your shared OneDrive called Q7 Program.exe. Download and run this program and you will see how it works.

b. Next write test data for this program.

c. Next code the program.

d. Test your code against the test data.

Q.8

a. Write the pseudocode algorithm for this program. Write a program which ***asks the user to enter*** a start and end number in the **main( )** function. The program must have the following error checking and appropriate error messages should be displayed: Both numbers entered by the user have to be greater than 0 and the second number needs to be greater than the first number entered. The start and end numbers do not need to be reentered if incorrect.

If valid numbers are entered the main() then calls a function called **addNumbers()** which accepts the start and end numbers as arguments. The addNumbers( ) function should add up all the numbers between the start and end number arguments (inclusive) and return the sum.

The function addNumbers() calculates the **sum** of these numbers and returns it to the main() function. The main function() displays the sum on the screen.

For example if the user enters the following:
Start number: 3
End number: 6

Total of these numbers is: 18

There is an executable program of this question on your shared OneDrive called Q8 Program.exe. Download and run this program and you will see how it works.

b. Next write test data for this program.
c. Next code the program.
d. Test your code against the test data.

Q.9

a. Write the pseudocode algorithm for this program. Write a function called **sumTotal()** which will **sum** a collection of number values. The user enters the number values. The number values must be between 5-12. An error message is displayed if the number inputted is outside this range. The user should continue to enter number values until a zero is entered. Your program should terminate when the user enters a zero value and the **sum** of numbers should then be displayed on the screen.

   There is an executable program of this question on your shared OneDrive called Q9 Program.exe. Download and run this program and you will see how it works.

b. Next write test data for this program.

c. Next code the program.

d. Test your code against the test data.

Q.10 Using a Menu

Write a simple menu driven program that will present the user with the following conversion menu (look up conversion rates on the internet):

1. Fahrenheit to Celcius
2. Inches to Centimetres
3. Exit

The program keeps presenting the user with options until the user enters the value 3 to exit.
Note: ***Switch*** statements and **do-while** loops are ideal for menus. Write the menu using a do-while loop.

Note: You should write a ***function*** for each of the conversions and call the function from the switch statement. For each of the conversions, the user should be asked to input a value. For example option 1 Fahrenheit to Celcius, the user should be asked to input the fahrenheit temperature and your program should work out the temperature in celcius and display it on the screen.

Q.11

a. Write the pseudocode algorithm for this program. Write a program which will ask the user to enter a number in the **main()** function. An error message should be displayed if zero or a negative number is entered and the user should be asked to re-enter the number.

It then passes this number as an argument to a function called **reverseDisplay().** The function reverseDisplay( ) should display every **third** number in reverse.

Example 1

```
Enter a number: 0
Error the number entered must be greater than zero – please reenter.
Enter a number: -5
Error the number entered must be greater than zero – please reenter.
Enter a number: 9
9       6       3       0
```

Example 2

```
Enter a number: 19
19      16      13      10      7       4       1
```

and so on.

Note: No negative numbers should be displayed on the screen.

There is an executable program of this question on your shared OneDrive called 11 Program.exe. Download and run this program and you will see how it works.

Modification
Modify the function **reverseDisplay()** to sum and calculate the average of the numbers displayed. It should return the average to the main() function and display it on the screen within the main() function.

Example 3

```
Enter a number: 9
9       6       3       0

The sum is 18
The average is 4.5
```

b. Next write test data for this program.
c. Next code the program.
d. Test your code against the test data.

Q.12

a. Write the pseudocode algorithm for this program. Write a function called **countOccurences()** which will **ask the user** how many numbers they wish to enter. An error message should be displayed if zero or a negative number is entered.

The function will then ask the user to enter that number of numbers. The numbers entered must be in the range 1 to 10.
Your program should contain **error checking**, for example if the user enters a number outside the range 1-10, the user should be asked to reenter the number.

It will then tell the user the highest and lowest number and how many times they have occurred. This information should be output after all the numbers have been entered.

For example if the following numbers are entered:
6 4 8 7 3 8 3 3

The highest number is 8 and occurs 2 times.
The lowest number is 3 and occurs 3 times.

Note: What should be output if the user enters all the same number?
3 3 3 3 3

There is an executable program of this question on your shared OneDrive called 12 Program.exe. Download and run this program and you will see how it works.

b. Next write test data for this program.
c. Next code the program.
d. Test your code against the test data.

Q.13

a. Write the pseudocode algorithm for this program. Write a function called **findOccurences()** which will ask the user how many numbers they wish to enter. The function will then read the integer values from the user and find and display the position of the **first** occurrence and the **last** occurrence of the number 9. Your function should print the position value of 0 (zero) if the number 9 is not found. The position is the sequence number of the data item 9. For example if the 8th data item is the only 9, the position value 8 should be displayed for the first and last occurrence.

9 9 4 3 9 8 7 6

b. Next write test data for this program.
c. Next code the program.
d. Test your code against the test data.

Q.14

a. Write the pseudocode algorithm for this program.
The **Fibonacci** series is the series of numbers:

1 1 2 3 5 8 13 ….

Each number (except the first two) is the sum of the previous two numbers. The first two numbers are 1 and 1. The series is supposed to govern growth in plants. Write a function called **fibonacciSeries()** which will calculate and display the first 20 Fibonacci numbers.

b. Next write test data for this program.
c. Next code the program.
d. Test your code against the test data.

Q.15

The program code below should do the following: Randomly generate 10 EVEN numbers using a loop. Use the ***rand( )*** function to generate the random numbers in the range 1 to 100. It should **display** and **sum** the even numbers.
Does it work correctly? If not can you fix the problem(s) in it?

```
int main()
{
       int step = 1;
       int checkEven = 0;
       int sum = 0;

       srand(time(nullptr));  // set the random number seed

       while (step <= 10)
       {
              int numberGuess = (rand() % 100) + 1;  // Generates a random no

              checkEven = numberGuess % 2; // check for an even number

              if (checkEven = 0) // if even
              {
                     std::cout << " " << numberGuess;
                     sum = sum + numberGuess;
                     step++;
              }// END IF

       } // end while

       std::cout << std::endl;
       std::cout << "The sum of even numbers is " << sum << std::endl;
       system("Pause");
       return 0;
}
```

Q.16

The program code below should do the following: Write a program which ***asks the user to enter*** a start and end number in the main( ) function. The program then calls a function called **addNumbers()** which accepts two arguments. The addNumbers( ) function should add up all the numbers between the start and end number arguments.

For example if the user enters the following:
Start number: 3
End number: 6
Total of these numbers is: 18

The program calculates the sum of these numbers and displays it on screen.
What does the following program do? Does it work correctly?
If not can you fix the problem(s) in it?

```
int main()
{
       int startNum = 0;
       int endNum = 0;

       std::cout << "Enter the start number : ";
       std::cin >> startNum;

       std::cout << "Enter the end number : ";
       std::cin >> endNum;

       addNumbers(startNum, endNum);

       std::cout << std::endl;
       system("Pause");
       return 0;
}

void addNumbers(int t_startNum, int t_endNum)
{
       int total = 0;

       std::cout << "The numbers are : ";

       while (t_startNum <= t_endNum)
       {
              std::cout << " " << t_startNum;  //write the number to the screen
              int total = total + t_startNum;   // sum the values
              t_startNum++;
       } // end while

       std::cout << std::endl;
       std::cout << "The sum is " << total;
}
```

Q.17

What does the following program do?
Does it work correctly?
If not can you fix the problem(s) in it?

```
int main()
{
        int count = 1;
        double sum = 0;
        int numberGuess = -1;
        int quantityNum = 0;
        double average = 0;
        int lowestNum = 0;
        int highestNum = 0;

        std::cout << std::endl << "Enter the number of numbers you want: ";
        std::cin >> quantityNum;


        while (count <= quantityNum)
        {
                std::cout << std::endl << "Please enter a number (1-10) : ";
                std::cin >> numberGuess;

                if (numberGuess >= 1 || numberGuess <= 10)
                {
                        if (numberGuess < lowestNum)
                        {
                                lowestNum = numberGuess;
                        }

                        if (numberGuess > highestNum)
                        {
                                highestNum = numberGuess;
                        }

                        sum = sum + numberGuess;

                }
                count++;
        } // end while

        average = sum / quantityNum;  // calculate the average number

        std::cout << "The lowest of the numbers is " << lowestNum << std::endl;
        std::cout << "The highest of the numbers is " << highestNum << std::endl;
        std::cout << "The sum of the numbers is " << sum << std::endl;
        std::cout << "The average of the numbers is " << average << std::endl;

        system("Pause");
        return 0;
}
```

Q.18

Calculations on a Sphere & Cylinder

Write a **pseudocode design** which will ask the user to enter the height and the radius of a vessel. The algorithm will then calculate and output the following values:

1. The volume of a sphere (i.e. $\frac{4}{3}\pi R^3$)
2. The surface area of a sphere (i.e. $4\pi R^2$)
3. The volume of a cylinder (i.e. $\pi R^2 H$)
4. The surface area of a cylinder (i.e. $2\pi R^2 + 2\pi RH$)

The program should continue to ask the user to enter height and radius values and display the volume, surface area etc until the user enters negative numbers for the height and radius. The program should then end.

The calculations for the volume, surface area, volume of a cyclinder and surface area of a cylinder should be done in **separate functions**. The radius should be passed as an argument to the functions.

Make sure to **test** that your output values are correct, by checking the answers on your calculator. Write the C++ code for this program.