



Programming Fundamentals (CS 1002)

Cybersecurity Department

Fall 2022 ASSIGNMENT # 3

Due Date: Friday, November 04, 2022 (11:59 pm)

Instructions

1. Assignments are to be done individually. You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own and you must understand each part of your code. You are encouraged to get help from the instructional staff through google classroom/Piazza.
2. Use appropriate data types and operations for each problem. You cannot use advanced topics not covered so far.
3. Your code must be **generic** i.e. it should work for different inputs where inputs are required.
4. The output should be properly displayed and well presented. **5% marks will be deducted in each question if appropriate comments and indentation** not done in source code.
5. **Plagiarism:** Plagiarism of any kind (copying from others, copying from the internet, etc) is not allowed. **If found plagiarized, you will be awarded zero marks** in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in the course.
6. **Submission Guidelines:** Dear students, we will be using auto-grading tools, so **failure to submit according to the below format would result in zero marks** in the relevant evaluation instrument.
 - a. Run and test your program on a lab machine before submission. If there is a **syntax error, zero marks** will be awarded in that specific question.
 - b. For each question in your assignment, make a separate .cpp file e.g. for question 1, make **ROLL-NUM_SECTION_q1.cpp**, and so on (e.g. **22i-0001_A_q1.cpp, 22i-0001_A_q2.cpp, 22i-0001_A_q3.cpp** and so on).
 - c. In every .cpp file that you create (every question), you must write your name, student-id, and assignment # on the top of the file in the comments.
 - d. Combine all your work (all questions files) in one folder. The folder must contain **only .cpp files** (no binaries, no exe files etc.,). If we unable to download your submission due to any reason you will be awarded zero mark.
 - e. Rename the folder as **ROLL-NUM_SECTION** (e.g. 22i-0001_A) and compress the folder as a zip file. (e.g. **22i-0001_A.zip**). Only zip file will be acceptable.
 - f. Submit the .zip file on Google Classroom within the deadline. Make sure that you have submitted the correct file.
 - g. Submission other than Google classroom (e.g. email etc.) will not be accepted.
 - h. The student is solely responsible to check the final zip files for issues like corrupt files, viruses in the file, mistakenly exe sent. If we cannot download the file from Google classroom due to any reason it will lead to zero marks in the assignment.
7. **Late submission:** **10% marks will be deducted for every hour of late submission**, i.e. assignments submitted 10 hours late will get zero marks. 10% will be deducted with the start of hour so both submissions on 10:01 and 10:59 will get same deduction of 10%.



- Using for loop write a C++ program get mean and average of the 5 scores of student. Also tell how many scores are more than 80.

```
Enter score of subject 1: 34
Enter score of subject 2: 105
Invalid score – score can be in the range 0-100
Enter score of subject 2: 75
Enter score of subject 3: 80
Enter score of subject 4: 78
Enter score of subject 5: 95
```

Mean of scores: **value**

Average of score: **value**

There are 2 scores which are more than 80.

- One interesting application of computers is drawing graphs and bar charts (sometimes called “histograms”). Write a program that reads five numbers (each between 1 and 30). After user input, your program should print a line containing that number of adjacent asterisks i.e. 5 lines of the size entered by the user. For example, if your program reads the number seven, it should print *********. Don’t use setw or setfill manipulators.

Moreover, your chart bars must be drawn in different colours chosen randomly. To learn how to change the colour of text, you can see guide “[Using Text Colors in your programs.docx](#)” available in the shared folder “[...Student Resources - All PF Course Material\Reference Material\Using Text Colors in your programs.docx](#)”

- Write for loop that produces the following output. Use nested while loops to capture the structure of the figure.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
\\!!!!!!!!!!!!!!!!!!!!!!!!!!//
\\\\\\!!!!!!!!!!!!!!!!!!!!//
\\\\\\\\\\!!!!!!!!!!!!!!//
\\\\\\\\\\\\\\!!!!!!!!!!//
\\\\\\\\\\\\\\\\\\!!!!!!//
\\\\\\\\\\\\\\\\\\\\\\!!!!//
```

- Write while loops to produce the following output:

```
##
# #
#  #
#   #
#    #
#     #
```



5. Write while loops to produce the following output:

```
1
22
333
4444
55555
```

6. Write for loops to produce the following output, with each line 40 characters wide:

```
-----
_ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^
1122334455667788990011223344556677889900
-----
```

7. It's common to print a rotating, increasing list of single-digit numbers at the start of a program's output as a visual guide to number the columns of the output to follow. With this in mind, write nested while loops to produce the following output, with each line 60 characters wide:

```
      |      |      |      |      |
123456789012345678901234567890123456789012345678901234567890
```

8. Write a program that produces the following output (with loops):

```
***** /////////////// *****
***** //////////////\ *****
**** //////////////\ ****
*** //////////////\ ***
** //////////////\ **
* //////////////\ *
```

9. Write a program that produces the following output (with loops, Remember Divide and Conquer):

```
+-----+
|  ^  ^  |
| ^  ^  |
| ^  ^  |
|  ^  ^  |
| ^  ^  |
+-----+
| v  v  |
| v  v  |
|  vv  |
| v  v  |
| v  v  |
|  vv  |
+-----+
```

10. Write a do - while loop that produces the following output:

$$\frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \frac{15}{16}, \frac{31}{32} \dots \dots \dots$$

11. Write a program that produces the following output (with loops):

12. Write a program that produces the following output (with loops):

```

      1
    1 2 1
  1 2 3 2 1
1 2 3 4 3 2 1
  1 2 3 4 5 4 3 2 1
    1 2 3 4 5 6 5 4 3 2 1
      1 2 3 4 5 6 7 6 5 4 3 2 1
        1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
          1 2 3 4 5 6 7 8
            1 2 3 4 5 6
              1 2 3 4
                1 2
                  1

```