

National University of Computer and Emerging Sciences



Programming Fundamentals CS188 Laboratory Manual

Course Instructor

Mirza Mubasher Baig

Lab Instructor(s)

Samia Akhter & Faraz Yousaf

Section

BDS-1A1 & A2

Semester


FALL 2021

FAST School of Computing

Department of Software Engineering

FAST-NU, Lahore, Pakistan

National University of Computer and Emerging Sciences, Lahore Campus

| | | | | |
|---|-----------------|--------------------------|---------------|--------------|
|  | Lab No 4 | | | |
| | Course Name: | Programming Fundamentals | Course Code: | CS 1002 |
| | Program: | BS(DS) | Semester: | Fall 2021 |
| | Duration: | 2.5 hours | Total Points: | 10 + 40 + 50 |
| | Lab Date: | Friday, October 15, 2021 | Weight | 4% |
| | Section: | BDS-1A | Page(s): | |

Instruction/Notes: Cheating during the lab will result in negative marks

Topics Covered: Repetition Structure

Use any of the IDEs including CodeBlocks, Visual Code, Visual Studio or the simple IDE available at <https://www.onlinegdb.com/> to write the programs

Submission Instructions:

1. Save all .cpp files according to the following naming convention
{ROLLNO}_{ACTIVITYNO}_{TASKNO}.cpp **FOR EXAMPLE.** 21L-XXXX_A01_P01.cpp, 21L-XXXX_A01_P02.cpp, 21L-XXXX_A02_P01.cpp
2. Now create a new folder according to the following naming convention
{ROLLNO}_{LABNO} e.g. 21L-XXXX_L02
3. Move all of your .cpp files to this newly created directory and compress it into a single file.
4. Submit this compressed file on Google Classroom.
5. You will get 10 Bonus point if you will follow these instructions correctly

Activity No 1

Problem No 1:

The following C++ code segments can be used to print **n** spaces or **n** * on the output device/stream

| Code segment to print spaces |
|--|
| <pre>for(int i = 0; i < n; i++) cout<<" ";</pre> |
| Code Segment to print * |
| <pre>for(int i = 0; i < n; i++) cout<<"* ";</pre> |

Use these code segments within a loop to create programs that can print the following patterns

| | |
|---|---|
| <p>Filled Square of height H</p> <p>The value of H will be given by the user</p> | <p>Sample output for H = 8</p> <pre>* *</pre> |
| <p>Hollow Square of height H</p> <p>The value of H will be given by the user</p> | <p>Sample output for H = 6</p> <pre>* * * * * * * * * * * * * * * * * *</pre> |

Filled Square of
height **H**

The value of H will
be given by the
user

Sample output for H = 8

```

      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * *
* * * * *
* * * * *
```

Hollow Triangle of
height **H**

The value of H will
be given by the
user

Sample output for H = 8

```

      *
     * *
    *   *
   *     *
  *       *
 *         *
*           *
* * * * *
* * * * *
```

Activity No 2.

Problem No 1:

Write a C++ program to prints the first **n** terms of the following **Tick-Tock** series. The value of n is taken as input.

The first 8 terms of the **Tick-Tock** series, i.e., for **n=8** are as follows:

$$2 - 5 + 4 - 10 + 6 - 15 + 8 - 20$$

Can you guess the pattern?

The odd terms are multiples of 2 and in increasing order (2, 4, 6, 8 etc.). The even terms are increasing multiples of 5 and in negative form. The terms are alternatively positive and negative. Nothing should be printed if the value of n is below 1.

For Example:

If the input is n=5, the program must print: **2 - 5 + 4 - 10 + 6**

If input is n=10, the program must print: **2 - 5 + 4 - 10 + 6 - 15 + 8 - 20 + 10 - 25**

Problem No 2:

Write a C++ program to prints the first **n** terms of the following Fibonacci series. The value of n is taken as input.

The first 10 terms of the **Fibonacci series**, i.e., for **n = 10** are as follows:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55$$

So the first two terms are 1 and 1 and every successive term is sum of the previous two terms.

For Example:

If the input is n=5, the program must print: **1, 1, 2, 3, 5.**

If input is n=10, the program must print: **1, 1, 2, 3, 5, 8, 13, 21, 34, 55.**