# Department of Computing

# School of Electrical Engineering and Computer Science

**CS-250: Data Structure and Algorithms**

**Class: BSCS 10C**

**Lab 07:  Movemin Problem, Version Control & Unit Testing**

**Date: 05th November, 2021**

**Time: 9:00 am – 11:50 am**

# Instructor: Prof. Dr. Faisal Shafait

# Lab Engineer: Mr. Aftab Farooq

# Lab 06: MoveMin Problem, Version Control & Unit Testing

**Introduction**

This lab is based on the concept of running time analysis, version control and unit testing.

**Objectives**

The purpose of this lab is make students practice the concept of version control and test cases.

**Description**

Follow the steps given in the lab tasks section and complete the lab tasks. Solutions of all tasks must be uploaded on LMS. No submission will be acceptable via email. Create a private repository called cs250-lab7 on bitbucket / GitHub. After completing each task, commit your code in the repository and attach a snapshot of the commit history along with the code in each task.

**Tools/Software Requirement**

Visual Studio C++

**Lab Tasks**

**Task 1:**

Write a C++ program that simulates the throw of a dice. It should take the input from a user about how many times the dice should be thrown and then generate a sequence of numbers.

* The output sequence may look like 4 1 6 3 2 …

LMS Upload:

* Source Code of the program
* Commit history screenshot

**Task 2:**

Consider the moveMin problem – all elements in the array are sorted except the last element, e.g.

3, 5, 12, 24, 25, 27, 15

* Implement the naïve solution (e.g. using bubble sort) using two nested loops

bool moveMin(vector<int> &in, vector<int> & out)

* Write test cases. Write a function bool testMoveMin()
  + It should generate a random array of integers in the range 1-100.
  + Store the generated array in a std::vector
  + Sort the vector using stl::sort()
  + Generate another random number and push it at the end of the array
* How do you create the test output data?
  + Copy the test vector into another vector and sort the new vector
* Run the test case
* Call the implemented function with the test input data and compare its result with the test output data

LMS Upload:

* Well commented source code
* Commit history screenshot

**Task 3:**

* Now implement the moveMin method using the single for loop
* Run the test cases to verify the correctness of your newly implemented faster algorithm
* Generate test cases of different sizes (10, 100, 1000, 10000, 100000, …) and note the difference in running times of both algorithms.
* Ask Google how to calculate running time of your C++ function
* Now run the test case with N=100000 at least one hundred times and compute
  + Best case running time
  + Worst case running time
  + Average case running time

**LMS Upload:**

* Well commented source code
* A plot of the running times of both algorithms
* A short report in Latex (both .tex and .pdf files) about the running time comparison of the faster and slower algorithms (including the plot) as well as the best / worst / average running times in the form of a table.

You can use overleaf for writing the report:  
<https://www.overleaf.com/>

**Lab Grading:**

|  |  |
| --- | --- |
| **Task** | **Marks** |
| Lab Viva/Quiz | 5 |
| Comments/ Indentation | 2 |
| Solution Document | 2 |
| Output Screen Shots | 1 |
| -- | -- |
| Total | 10 |

**Deliverables**

This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems discuss it by emailing it to [aftab.farooq@seecs.edu.pk](mailto:aftab.farooq@seecs.edu.pk).

**Note:** Students are required to upload the lab on LMS before deadline.

Use proper indentation and comments. Lack of comments and indentation will result in deduction of marks.