# Department of Computing

**CS213: Advanced Programming**

**Class: BSCS – 5C**

# Lab 2: Understanding Unit Testing and Unit Test frameworks

**CLO4: Apply various code management tools and techniques in modern languages.**

**Date: September 22, 2017**

# Time: Friday (14:00 – 17:00)

# Instructor: Shamyl Bin Mansoor

# 

# Lab 2: Understanding Unit Testing and Unit Test frameworks

## Introduction

In this lab we will learn how to use a unit testing framework and write unit tests for codes that we develop for a few problems.

* Unit testing of software applications is done during the development (coding) of an application.
* The objective of unit testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure
* The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. Unit testing is usually performed by the developer

## Objectives

* Understand how to use a framework for writing unit tests
* Learning to write unit tests using Visual Studio’s CppUnitTest Framework

## Tools/Software Requirement

* Solutions can be developed in any language preferable Python, C++ or Java. However, ensure that your code is well documented to be easily understood.
* **Whatever language you choose to develop your programs, you have to use the appropriate Unit Testing framework. For C++ use CppUnitTest, for Python use unittest2 framework or similar and for Java use JUnit.**

**Description**

Each student must, individually build the complete application on their own. Students must upload their solutions on LMS to qualify for evaluation. The following reasons can lead to losing marks

* Any exceptions or errors leading to non-execution of submitted code.
* Failure to upload the solution on LMS.
* Failure to submit original code.
* Failure to use GitHub for the code.
* Failure to explain the submission, during viva.

**Lab Task**

1. Create a class library for recursively calculating the factorial of a number.
2. Develop unit tests for this class library.
3. Develop a Class library for Stacks. Do not use any builtin library, develop your own from scratch.
4. Write Unit Tests for it

## Deliverables

* Each submission is individual with the following composition:
  + Source Code
  + Documentation (Comments)
  + Link to the public repository preferably on GitHub
* Convert your submission files into a zip folder and name it as given below, finally upload the zip folder to LMS.
  + Name – Registration No. – Section

## Grade Criteria

This lab is graded. Min marks: 0. Max marks: 10.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Minimum** | **Maximum** |
| Learning to use Unit Test Framework successfully | 0 | 1 |
| Code clarity with clean, formatted and commented code. | 0 | 1 |
| Class for Recursive Factorial | 0 | 1 |
| Unit Tests for Recursive Factorial function | 0 | 2 |
| Class for Stacks | 0 | 1 |
| Unit Tests for Stacks Class | 0 | 2 |
| Viva | 0 | 2 |
| **Total** | **0** | **10** |

**Links for Learning:**

<https://www.codeproject.com/Tips/1085171/How-To-Do-Unit-Testing-with-Cplusplus-in-Visual-St>