# Department of Computing

**CS213: Advanced Programming**

**Class: BSCS – 4AB**

# Lab 6: Build Your Own PC(BYOPC)

**Date: November 3rd, 2016**

**Time: Thursday (10:00 AM – 01:00 PM & 02:00 PM - 05:00 PM)**

# Instructor: Fahad Ahmed Satti

# Lab 6: Build Your Own PC (BYOPC)

## Introduction

In this lab, students will simulate an automated assembly line for building a large number of PCs. Take a look at the description section to understand the problem scenario.

## Objectives

After performing this lab students will be able to understand:

* Simulation
* Recursion
* Dynamic Programming

## Tools/Software Requirement

* Preferred Programming Language: C++, Java or Python
* Any IDE
* SVN – Github

## Description

A new computer building company wants to automate the process of PC building by using a 3 lane assembly line. The assembly line has 3 automated transfer belts where the workers place a chassis selected by a particular user. The chassis then goes through n stations. At each station a new component is added to the chassis in some random amount of time. A finished PC is taken out of the automated belt by a worker and placed in a box.

All stations at the same index, perform the same operation, however some random station can break at any time or add a component incorrectly, requiring human intervention. As a result the time spent by the chassis on each station can be different. If a node breaks down, a worker marks it as not useable and manually changes the automated belt for the chassis.

You have been asked by the company to simulate some least time taking paths for them, without actually peeking into the system. The finance manager wants to utilize all 3 lanes by building 3 PCs simultaneously. The workers are adamant; only 1 PC can be built at a time. The administration would like to take a middle way. The CEO however wants a proof of concept first. He wants a software simulation that can prove to him the best paths with least amount of time to take, if building 1, 2, or 3 PCs simultaneously. You will have to provide 3 different solutions utilizing recursion, recursion with memorization and dynamic programming.

## Lab Task

Your task will be to implement the following:

1. Simulate a 3-lane assembly line via recursive solution.
2. Simulate a 3-lane assembly line via recursive solution with memoization.
3. Simulate a 3-lane assembly line via dynamic programming (iterative).
4. Find out best paths when1, 2, or 3 PCs are simultaneously placed on the assembly line.
5. Profile your simulations for large n(1000,10000, and 100000).
6. Unit tests to evaluate your code.
7. Using a Version Control System (VCS) to manage your solutions.

## Deliverables

Following the guidelines set out in your course outline, your submission must include the following:

1. Unit tests
2. A description document with separate sections on Introduction, How to run your application, Link to public GitHub repo with your submission and any other data.
3. Original Source Code

Convert your submission files to a zip folder and name it as given below and upload the zip folder on LMS.

**Name – Registration No. – Section**

## Grade Criteria

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

|  |  |  |
| --- | --- | --- |
| Activity | Minimum | Maximum |
| Documentation with clearly defined understanding of the lab task and approach | Fail | Pass |
| Code clarity | 0 | 1 |
| Github (SVN) | 0 | 1 |
| Unit Tests | 0 | 1 |
| Viva | 0 | 3 |
| Lab Task | 0 | 4 |