# Department of Computing

**CS 213: Advanced Programming**

**Class: BSCS 5 AB**

# Lab 3: Functional Problem Set

**Date: October 4th, 2017**

**Time: Thursday (10:00-12:50 & 14:00 – 16:50)**

# Instructor: Fahad Ahmed Satti

# 

# Lab 3: Functional Problem Set

## Introduction

In this open ended lab the students have to use functional paradigm in Scala and provide solution to the problem set.

## Objectives

* Familiarize yourself with the use of Functional Programming Paradigm in Scala.
* Write tests to show your solutions are correct.

## Tools/Software Requirement

* Solutions should be made using Functional programming paradigm in Scala.
* You can take help from internet but remember **no plagiarism.**

**Description**

For each problem defined in the Lab Task section provide a solution using functional programming paradigm in Scala. The difficulty level of each problem is indicated by an asterisk (\*). Easy problems are marked with an \*, moderate ones are \*\*, and relatively difficult ones are marked as \*\*\*. For this problem set, you are not allowed to use any built-in functions.

**Lab Task**

1. \*Find out the last Nth element in a randomly generated list.
   1. scala> lastNth (3,List(1,1,2,2,3,3,5,5,8,9,10))
   2. res0: Int = 8
2. \*Find out the last Nth element in a randomly generated list using a tail recursive solution.
   1. scala> lastNthRecursive(3,List(3,3,5,5,8,9,10))
   2. res0: Int = 8
3. \*Determine whether a given integer number is prime or not
   1. scala> isPrime(9)
   2. res3: Boolean = false
4. scala> isPrime(11)
5. res4: Boolean = true
6. \*\*Determine whether two positive integer numbers are coprime.( Two numbers are coprime if their greatest common divisor equals 1)
7. \*\*Define XOR operation for two logical expressions.
8. \*\*Define AND operation for two logical expressions.
9. \*Define NAND operation for two logical expressions.
10. \*\*\*Given 5 random numbers (between 1 and 100), determine the mathematical operations (sum, subtract, product, divide) which can produce a number X, where each number is used only once. Example: How can 3, 7, 6, 8, 1 produce 348? (((8 \* 7) + 3) -1) \*6 = 348

## Deliverables

* Each submission is individual with the following composition:
  + Source Code
  + Tests
  + Documentation(Introduction, Approach, Design, How to Run and Analysis)
  + Link to the public repo 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 will be graded in the following way:

|  |  |  |
| --- | --- | --- |
| Task | Min Marks | Max Marks |
| 1. lastNth | 0 | 2 |
| 1. lastNthRecursive | 0 | 2 |
| 1. isPrime | 0 | 2 |
| 1. isCoPrime | 0 | 3 |
| 1. xor | 0 | 3 |
| 1. and | 0 | 3 |
| 1. nand | 0 | 2 |
| 1. missingOperators | 0 | 7 |
| **Total** | | 24 |