National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

## Department of Computing

# CS213: Advanced Programming

## Class: BESE – 4B

### Lab 11: Functional programming in Scheme

## Date: 23rd May, 2016

## Time: 10:00 AM to 12:50 PM

### Instructor: Mr. Numair Khan

# Lab 11: Scheme

## Introduction

Functional programming languages are practical implementations of lambda calculus, which we have been studying for the past few lectures. In this lab we are going to learn to use Scheme to implement two imperative programming constructs – if-else statements and for loops.

## Objectives

After performing this lab students will be able to:

- Write code in Scheme
- Translate imperative code to its functional variant

## Tools/Software Requirement

- MIT/GNU Scheme

## Deadline

- The deadline for submitting Lab 11 is 23:55PM on Tuesday, 24th May 2016.

**If-else statements**

As discussed in class, if-else statements can be implemented using our functional definition of true and false:

True $\equiv \lambda ab.a$ False $\equiv \lambda ab.b$

The format of the if-else statement then becomes:

If-else ≡ (Lambda term that reduces to True/False) (If branch) (Else branch)

Bear in mind, that **everything is a function in pure lambda calculus**, including the if-else branches.

**For loops**

While we do not have loops in lambda calculus, we have learnt to implement recursion using the Huskell Curry Fixed-point Combinator (Y-combinator).

$Y \equiv \lambda f. (\lambda x. f(xx)) (\lambda x. f(xx))$

To implement recursion, pass your recursive function as an argument to the Y-combinator:

Recursive implementation of $F \equiv Y\ F$

Remember to include a base condition in F.

## Lab Task

Implement the following C code in Scheme. (You may use numeric constants (1, 2, 3….) instead of Church numerals).

```
int aCompletelyUselessFunctionCreatedOnlyToMakeYourLifeMiserable(int m, int n) {
     if( m == 0 || m < n) {
        int sum = 0;
        for(int i = 0; i < n; i++) {
            sum += i;
        }
     }
     else {
        return n + m;
     }
}
```

## Deliverables

A link to the Git repository where you will upload your Scheme code.