



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

Department of Computing

CS213: Advanced Programming

Class: BESE – 4B

Lab 10: Scheme

Date: 16th May, 2016

Time: 10:00 AM to 12:50 PM

Instructor: Mr. Numair Khan



Lab 10: 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, a popular functional programming language. However, as already mentioned, all functional languages have a common base – lambda calculus. Therefore, once you understand the basics, learning a new functional programming language of your choice (Haskell, ML, Scala) will be as easy as saying, “Damn you, Alonso Church!”

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 10 is 23:55PM on Tuesday, 17th May 2016.



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

Lab Task 1 – Download and Install GNU/MIT Scheme

Download and install GNU/MIT Scheme.

Lab Task 2 – Functions

Scheme does allow you to implement impure lambda calculus. That is, functions like `+`, `*`, `<`, etc are predefined as constants. However, for the current lab, we will be avoiding all impure constructs of the language. Your task is to define the following functions as we studied in class (consult the lecture slides if you need a refresher):

1. Add
2. Subtract
3. And
4. Or
5. Not
6. LEQ
7. GEQ

I asked a guy named Google, and he told me this is a good resource for learning Scheme: https://www.shido.info/lisp/idx_scm_e.html. Read the first four chapters (up to “Defining Functions”).

Deliverables

A text file containing the address of the GIT repository to which you will upload your scheme code.