

# Lab-1: Monday, Feb. 14, 2005

## Introduction to Scheme

In this first lab, you will be learning how to invoke Scheme, gain experience with defining and executing Scheme procedures, and learn how to submit labs and assignments.

### Getting Started

1. Log onto the machine in lab. Click on **Applications** at the top of the screen, then select **Programming**, and then **xterm** on **Strauss** in the drop-down menus that appear.
2. You are now logged into **strauss**. You can execute the command **chdgrp** to see what account your session will be charged to. If the account is not 2081, then execute the following command to have this session charged to our project:

```
newgrp 2081
```

3. If you do not have a `.localalias` file in your directory, then execute the following:

```
cp ~carberrry/.localalias .localalias
```

If you already have a `.localalias` file in your directory, then add the following lines to the end of it:

```
setenv PATH /opt/lib/plt/bin:$PATH
setenv MANPATH /opt/lib/plt/man:$MANPATH
setenv LD_LIBRARY_PATH /opt/sfw/lib
```

These are the paths needed for Scheme and its library.

4. Log out and then log back into **strauss** in order to establish the new paths.
5. (If the following fails to invoke Scheme, please ask your TA — you may need to reset some other files in your directory, and your TA will give you a handout describing what to do.) To invoke Scheme, execute the following:

```
drscheme
```

The DrScheme logo will appear.

- (a) The first time that you do this, another window will appear with the heading “Welcome to DrScheme” and with “Interact with DrScheme in English” selected. Click on the **Next** button in the lower right corner.
- (b) The next screen that appears says that you can check for updates. Ignore this and click on the **Next** button in the lower right corner.
- (c) The next screen asks you to select a language, with “Beginning Student” highlighted. Instead, under **Professional Languages**, click on **PLT** and then click on **Graphical (MrEd, includes MzScheme)**. Then click on the **Finish** button in the lower right corner.

6. A new window will appear consisting of two parts: the Definition window and the Interpreter window. Resize the window so that it is larger and easier to work with.
7. In the DrScheme environment, you will use a text-editing system which incorporates an editor closely resembling the widely used editor Emacs. Click on the “Help” button (at the top of the screen) and then select **Help Desk** to bring up the **PLT Help Desk**. Note that the Help Desk window is actually a web browser.
  - (a) Click on **Help Desk** and read through the information that appears; it tells you how to get help.
  - (b) Click on **Rewind** at the top of the Help Desk screen to get back to the main **PLT Help Desk** page.
  - (c) Click on **Software**, then on **DrScheme**, then on **Interface Essentials**.
  - (d) If necessary, resize the window so that is larger and more readable. Sections 2.1 through 2.4 describe the interface and you might want to examine it later. But you should be aware that you can get a lot of help from these Help Desk pages.

## Definition and Evaluation

1. In the interpreter buffer, type each of the following expressions, ending each with a carriage return. Is what is printed what you expected? If not, ask your TA to explain.

```
(+ (* 3 5 2) 4)
(* (- 7 3 2) (+ (* 2 4) (* 3 2)))
(define a (* 4 2))
(+ a (* 4 a))
```

2. Now go to the definitions window, and enter the following procedure for computing n-factorial:

```
(define n-factorial (lambda (n)
  (cond ((= n 0) 1)
        (else (* n (n-factorial (- n 1))))))
```

Notice that each time you hit carriage-return, the editor moves over to properly indent the next line according to the depth of the parentheses. It is **essential** that your procedures be properly indented so that they are understandable; otherwise it will be very difficult for you to decipher what they are doing and very difficult (if not impossible) to debug them. Points will be deducted on assignments for procedures that are not properly indented.

3. Once you have entered the definition for n-factorial in the definitions window, click on the **Check Syntax** box at the top of the screen. If there are errors in the syntax (such as missing parentheses), an error message will appear. If you typed the definition correctly, there should not be any error messages; if there are, make the appropriate corrections to your definition.
4. Click on the **Execute** button at the top of the screen. Recall that this *executes* the expressions in the definition window, so that now the n-factorial procedure is defined and can be used in the Interpreter window.
5. Move to the Interpreter window and enter the following expression:

(n-factorial 5)

Did you get the correct answer? You should have, as long as you entered the definition of n-factorial correctly.

6. You can remove n-factorial from the definition buffer by highlighting it via the mouse, then clicking on the **Edit** button at the top of the screen and selecting **Cut** from the drop-down menu that appears. Note that there are also **Copy** and **Paste** options.
7. Copy the file `~carberry/fib.scm` into your directory.
8. Click on the **File** button at the top of the screen and then select **Open** on the drop-down menu that appears. A new window will appear; select the file `fib.scm` and click on **OK** in the bottom right corner.
9. A new DrScheme window will appear. Click on **Execute** and the definition will be executed and the Interpreter buffer will appear.
10. Click on **Check Syntax**. An error message appears, indicating that there is a syntax error. In this case, the error is in the last line — there is a missing left parenthesis. Find the error and correct it. (**Look carefully at the procedure before you make changes — do not just start adding parentheses at random.**)
11. Once you have corrected the error, you will need to click **Execute** again. Then in the Interpreter window, enter the following expression:

(fib 6)

Do you get the correct answer? You should as long as you corrected the syntax error properly without putting other errors into the procedure definition. Once you get the procedure working correctly, you should save its definition in a file by clicking on the **File** button at the top of the screen, selecting either **Save Definitions** or **Save Definitions As**, and giving a file name — note that the second time you use **Save Definitions**, it will automatically use the same file name as the first time.

12. Now it is time to try constructing your own procedure definition. Define a procedure (**sum-prod x y z**) that returns the sum of x, y, and z if x is positive, returns the product of x, y, and z if x is negative, and returns the product of y and z if x is zero. Test your procedure for several values of x, y, and z. Once you have it working, save the procedure definition.
13. Now let's try submitting an assignment. Put your definitions for the corrected procedure in a single file called `lab1.scm`. Then do the following:
  - (a) Using a web browser, go to the web stie for your TA — be very careful to choose the correct web site, or else your TA will not receive your submission and you will not receive credit for it:

<u>Sections</u>	<u>TA</u>	<u>Web Address</u>
10, 11	Mike Smith	<a href="http://www.cis.udel.edu/~mismith/cisc280/">http://www.cis.udel.edu/~mismith/cisc280/</a>
12, 13	Manabu Torii	<a href="http://www.cis.udel.edu/~torii/cisc280/">http://www.cis.udel.edu/~torii/cisc280/</a>

A submission form will appear.

- i. Enter your UDel email address in the space provided.
  - ii. Select your correct section from the drop-down section menu.
  - iii. Click on the button to the left of **Lab**, then select **Lab 1** from the drop-down menu that appears.
  - iv. Paste your procedure definitions from file `lab1.scm` into the window labelled **Program**. The file must include **NOTHING** extra, only the procedure definitions that are to be tested.
  - v. Then click on the **Submit** button at the bottom of the page.
  - vi. You will quickly receive a reply from the submission system telling you whether your procedures executed correctly on our test data. You should click on **Confirm** if you are satisfied and want this to be your official submission; click **Cancel** if you are not satisfied with it and want to revise it. Note that at this point, you have not finished the lab so not all the procedures have been tested; thus you will get an error message.
- (b) Now try defining a recursive procedure (`sum-squares x y`) that returns the sum of the squares of all integers between  $x$  and  $y$  inclusive. You may assume that  $x$  and  $y$  are both positive and that  $x \leq y$ .
- (c) Execute your procedure `sum-squares` for various values of  $x$  and  $y$ . Once you believe it is working properly, put all of your procedures (`fib`, `sum-prod`, and `sum-squares`) into a single file. Go to the submission window, and submit the procedure definitions. If you receive a **PASS** from the submission system, click on **Confirm** to send them to your TA — Congratulations, you have completed lab-1. otherwise, try to figure out what is wrong, correct the procedures, and resubmit them.