## **CSE 240 Homework 10, Fall 2015 (50 points)**

Due Saturday, November 7, 2015 at 11:59PM, plus a 24-Hour grace period

### Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures. By the end of the assignment, you should have

* strong concept functional paradigm.
* strong concept Dr. Racket.
* Understood names and procedures

**Reading**: Text Chapter 4 and course notes (slides). This is a complete new language, and you need to spent more time to read and to program.

**Preparation**: Complete the multiple choice questions in the textbook exercise section. The answer keys can be found in the course Web site. These exercises can help you prepare for your weekly quiz and the exam. You are encourage to read the other exercise questions and make sure you understand these questions in the textbook exercise section, which can help you better understand what materials are expected to understand after the lectures and homework on each chapter.

You are expected to do the majority of the assignment outside the class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or the TA during their office hours.

You are encouraged to ask and answer questions on the course discussion board. However, **do not share your answers and code** in the course discussion board.

**Programming Environment Installation**: Download DrRocket Scheme from:

http://racket-lang.org/download/

Install the latest version. There are Windows, Mac, and Linux version available! You must choose – R5RS, which is a professional edition Notice that the book is based on R5RS (professional edition). If you choose the Advanced Student version, many features, such as the pair data structure and its operations are not supported.

Start the program and type your program in the upper window, as shown in the following diagram.



### Programming Exercise (50 points)

In this assignment, you will be learning Scheme through the use of Dr. Racket. We would like to start with some basic concepts; trying to under prefix notation and the use procedure in Scheme. You will also implement a few simple procedures. You may only use the procedures shown in the text and slides - not any of the additional library procedures in Scheme.

1. Using DrRacket Scheme compute the following expressions. [5 points]

* 3 + 5 + 7
* 3 + 3 + 5 + 7
* (3 \* 2) + (3 \* 8 \* 8)
* (3 \* 2) + (9 \* 9) + (18 / 3)
* (2 + 3 + 6 + 8 + 10 + 12 + 14) / (2 + 3 + 6)

In the following questions, you must use comment to explain the function of each procedure that you defined. Use comments to mark the question numbers too. Make sure that the entire file is executable.

1. Bind (define) the value 3 to its’ English text – “three”. Then replace the value 3 with the English text for all the expressions in question 1. Make sure the binding takes (definitions) place at the beginning of your definition before the expressions. [5 points]
2. Use *define* to bind base and height to two values and then define a procedure called “RectArea” that will accept a base and a height; in return, it will compute the area of a rectangle. [5 points]
3. Define a procedure called “RectVol” that will take a base, a height, and a depth; in return, it will compute the volume of a rectangle. **For the parameter depth, you must use (read) to provide its value.** Furthermore, you must use the RectArea procedure defined in question 3. You must print (display) the return value RectArea in the program. Note, the return value of RectVol will be printed automatically. Hint: You need to define a printing procedure that can return a value, as discussed in the lecture. The pseudo code in **infix** style is given as follows: [10 points]

RectVol (base, height):= (printVal (RecrArea (base, height)) \* read())

1. Define a procedure called “DiffVolume” that will take two rectangles (base, height, and depth); in return it will compute the difference in volume between the 2 rectangles. You must use RectVol procedure defined in question 4. The pseudo code in **infix** style is given as follows: [10 points]

DiffVolume := RectVol (base1, height1) - RectVol (base2, height2)

1. Reimplement the procedure RectArea and call it “RectAreaLet”. In this procedure, you must use *let*-form to bind base and height to two values, instead of using *define*. [10 points]

1. Reimplement the procedure RectVol and call it “RectVolLet”. In this procedure, you must use *let*-form to bind base and height to two values, instead of using *define*. The depth will be entered using (read). [5 points]

\*\* You will submit in this assignment in single executable file (.rkt).

You must write your test cases to test each of your procedures by given the necessary input values. Validate that your procedures generated the correct values. You do not have to submit the test cases. We will use our own test cases to test your procedures.

### Grading of Programming Assignment

The TA will grade your program following these steps:

(1) The TA will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

(2) Compile the code. If it does not compile, 20% of the points given in (1) will be deducted. For example, if you are given 20 points in step (1), your points will become 16 if the program fails to compile.

(3) If the code passes the compilation, the TA will execute and test the code. If, for any reason, the program gives an incorrect output or crashes for any input, 10% of the points given in (1) will be deducted.

### What to Submit?

You are required to submit your solutions in a compressed format (.zip). Zip all files into a single zip file. Make sure your compressed file is labeled correctly - lastname\_firstname.zip.

For this home assignment, the compressed file MUST contain the following:

Hw10.rkt (Scheme program)

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded, even if the assignment is submitted late.

### Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

### Late submission deduction policy

* No penalty for late submissions that are received within 24 hours after the deadline;
* 10% grade deduction for every day it is late after the grace period;
* No late submission after Tuesday at 11:59PM.

### Academic Integrity and Honor Code.

You are encouraged to cooperate in study group on learning the course materials. However, you may not cooperate on preparing the individual assignments. Anything that you turn in must be your own work: You must write up your own solution with your own understanding. If you use an idea that is found in a book or from other sources, or that was developed by someone else or jointly with some group, make sure you acknowledge the source and/or the names of the persons in the write-up for each problem. When you help your peers, you should never show your work to them. All assignment questions must be asked in the course discussion board. Asking assignment questions or making your assignment available in the public websites before the assignment due will be considered cheating.

The instructor and the TA will **CAREFULLY** check any possible proliferation or plagiarism. We will use the document/program comparison tools like MOSS (Measure Of Software Similarity: http://moss.stanford.edu/) to check any assignment that you submitted for grading. The Ira A. Fulton Schools of Engineering expect all students to adhere to ASU's policy on Academic Dishonesty. These policies can be found in the Code of Student Conduct:

http://www.asu.edu/studentaffairs/studentlife/judicial/academic\_integrity.htm

ALL cases of cheating or plagiarism will be handed to the Dean's office. Penalties include a failing grade in the class, a note on your official transcript that shows you were punished for cheating, suspension, expulsion and revocation of already awarded degrees.