Instruction Manual of Auto-Grader (Racket)

All the students should have defined a global variable called \*name\* in their own assignment file, and you can give this information about the assignment in class. The code should look like:

(define “student’s name” “my name”).

The instructor can make a new file called sample-test.rkt which they should code in the specific tests and the make-asmt struct.

First, the instructor should define individual tests and use the make-test function with the following inputs: function test name as string, designated points per test as a list, the number of max points, the name of the function as a symbol, a list of lists of inputs, the predicate, and a solution function that the instructor provides. The instructor can implement any predicate they deem necessary for the given problem.

Then, the instructor should define my-asmt with the make-asmt struct with the following inputs: assignment name as a string, the class name as a string, and the test structs as a list.

An example is given below:

(define test1

(make-test "FACTORIAL FUNCTION"

(2 2 2 2) ;; 2 point per each test

10 ;; number of max points

'facty ;; name of function (a symbol)

'(0 1 2 3 8 10) ;; list of inputs

equal? ;; predicate used for checking

(letrec ((facty-soln (lambda (n) ;; solution function

(if (<= n 1)

1

(\* n (facty-soln (- n 1)))))))

facty-soln)))

(define my-asmt

(make-asmt "ASMT-0" "CMPU-101" "4-21-2018" (list test-1 test-2)))

Transitioning to the grader.rkt file, the instructor should type in the Interaction window:

(grade-asmt "asmtN" "sample-test.rkt") with N being the number of the assignment.

The auto-grader will fetch contents of TEST data structure and run the student’s solution against function solution. If answers do not match, student will be awarded 0 points; if not, they will get however many points assigned.

After program is ran, the statement that displays is:

Filename: trav-asmt0.rkt

-------------------

STUDENT: Jerry

ASMT-INFO: ASMT-0, CMPU-101, DATE: 4-21-2018

-------------------

Test FACTY

| Expected Output | Student Output | Points

--------------------------------------------------------------------------

input(s): 1 | 1 | 2 | 0

input(s): 2 | 2 | 4 | 0

input(s): 5 | 120 | 240 | 0

input(s): 6 | 720 | 1440 | 0

input(s): 0 | 1 | 2 | 0

-------------------

SUBTOTAL: 0

-------------------

Test REV\_LISTY

| Expected Output | Student Output | Points

--------------------------------------------------------------------------

input(s): (1 2 3 4 6 | (12 11 10 9 8 7 | (12 11 10 9 8 7 | 2

input(s): () | () | () | 2

input(s): (5 4 3) | (3 4 5) | (3 4 5) | 2

input(s): (6) | (6) | (6) | 2

-------------------

SUBTOTAL: 8

-------------------

TOTAL SCORE: 8/18