Grading Guidelines for Assignments 1 and 2 (December 10)

Assignment 1 (Total Marks = 10):

- Output (binary marking: either 1 or 0 for each test case): 3 marks
    - Test case 1: point inside rectangle – should print inside
    - Test case 2: point outside rectangle – should print outside
    - Test case 3: Point on one side of rectangle, should say outside
    - TAs should coordinate to create the test cases, all programs to be tested against same test cases
- Code (use your judgement for partial marks): 7 marks
    - Reading in a and b in a single scanf – 1 mark
    - Reading in c and d in a single scanf – 1 mark
    - All four corners computed and printed correctly in correct order – 2 marks
    - If-else statement for checking point is inside or outside – 3 marks

Assignment 2 (Total Marks = 10):

- Output (binary marking: either 1 or 0 for each test case): 3 marks
    - Test Case 1: complex roots (should just print a message)
    - Test case 2: real roots, unequal (should print both roots)
    - Test case 3: real roots, equal (my print only once or twice, should also print the message. Maybe can give 0.5 if roots printed but no message)
    - TAs should coordinate to create the test cases, all programs to be tested against same test cases
- Code (use your judgement for partial marks): 7 marks
    - Reading in a, b, c with proper format string (ok if they use any double instead of float, and %f instead of %lf) – 1 mark
    - Checking and printing message for complex roots – 2 marks
    - Checking and printing real, unequal roots – 2 marks
    - Checking and printing real, equal roots – 2 marks
    - The codes for the last three parts will vary. Some have computed the expression in steps, some have just printed the roots directly. Give credit for all as long as they have these parts. However, deduct 1 if they have taken a square root on the discriminant directly before checking if < 0 or not.

For partial marking, use resolution of 0.5.

While grading the code, general idea is that ones who are close to correct should get almost full marks in code. If their programs did not compile or run for some small mistakes, they have already lost in output. Use your judgement as to how much to deduct, they can always ask you later.

For any marks deducted, please write some comments as to why it is so.

You may give any other comments to improve them as you see fit.