

Evaluation of Programs for the ITiCSE Working Group on Programming Skill  
Assessment  
Michael McCracken  
5/9/01

Since this is a programming exercise intended to evaluate the programming skills of the subjects, the evaluation will be focused on skills. Included below is an outline of the evaluation criteria that I hope can be followed without disturbing your normal grading process but will give us reasonably consistent evaluations.

1. Execution - Does the program execute without error in its initial form. That's a fancy way of saying does it compile without error, and when you invoke it, the program doesn't dump core or equivalently fail.
2. Verification - Does the program correctly produce answers to the data set you gave it.
  - The programs should allow for multiple entries of different problems. In other words, it should clear the data structure properly between problems.
  - The program should terminate correctly (entering the quit command should terminate the program).
  - The data sets to test the programs should be representative of the types of problems you would expect to calculate with a calculator. (Note, this will likely produce inconsistencies in evaluation, so I will attach some sample problems. They are not meant to be inclusive, but if we test to this minimum set, at least we have a benchmark). If you add test cases, that would be great. If the group wants we can run programs on your test cases before or at the conference. If you have or are going to develop your own test cases, my criteria was as follows:
    - Simple expressions to more complicated
    - Mixed mode arithmetic
    - Error cases (which should be detected)
    - Multiple operators
    - Typical operator errors (divide by zero, empty parentheses, multiple operators (where illegal)
    - Bad operators
    - Overflow cases (I might have failed at that one)
  - The program should react properly to erroneous inputs.
3. Validation - Does the program represent the calculator type you asked the student to build and is it a calculator?
4. Style - Is the program written to the standards that you expect your students to follow? This includes naming conventions, indentation, etc.

The marking system will be as follows:

Execution - 30 marks

Verification - 60 marks (The testing part (running against data sets) should constitute 30 of the 60 marks. The other three should be allocated 10 marks each.

Validation - 10 marks

Style - 10 marks

The points add up to 110 marks. The reason is for us to make the style measure optional in the final analysis. You may have no style requirements in your

