567 Project 1: Triangle Testing

**Tools:**

For your team, you need to get a configuration management, test mgmt, and bug tracking system up and working. You also need to play around with them, use them, and understand what they can do.

In the past, students have spent a lot of time finding and loading tools. Based on student suggestions, we’ve now got a tutorial for test mgmt and bug tracking systems on how to install for either a windowsXP or linux environment. Also, there is a link to a web site on configuration mgmt. You need to get a config mgmt tool up and working – you can select your own. Subversion tends to be popular with students. Look under tools on elearn for more info.

For the tutorials, if you find problems, please let brian woo ([bwoo@stevens.edu](mailto:bwoo@stevens.edu)) know. He is a part time student (with a full time job) and will help you if he can, but it is your responsibility is to get these up and working for your team.

Project:

For this week, all the teams will work on the same program, but we will be doing a software engineering experiment with different methodologies. The instructor will assign your team to one of the methodologies. Although you are working together as a team, each team member should write the program themselves.

The Program:

Each team member needs to write a program that accepts three numbers, a, b, and c, interprets those numbers as the lengths of the sides of a triangle, and outputs the type of triangle. (Equilateral, Isosceles, Scalene, Right). [Don’t worry too much about this being perfect the first time. This is not a programming competition! ]

The Methodologies

Methodology 1: Just Write it!:

For week one, write the program and then run one or two tests on it (no more). Put the code in your configuration mgmt system, and put your test cases (and results) in your test mgmt system. If you find any bugs, put them into your bug system, and then fix the bugs.

Methodology 2: Requirements First

For week one, write a set of requirements, review them among yourselves, put them under change control, then write the program and then run one or two tests on it (no more). Put the requirements and code in your configuration mgmt system, and put your test cases that you ran in your test mgmt system. If you find any bugs, put them into your bug system, and then fix the bugs.

Methodology 3: Test Cases First

For week one, write your set of test cases first (with expected results) and put them in your test mgmt system. Then write the program, and then run one or two tests on it (no more). Put the code in your configuration mgmt system, and put your test cases (and results) in your test mgmt system. If you find any bugs, put them into your bug system, and then fix the bugs.

Project Report should include:

1. Standard Info for all projects:
   * + The assignment (write it out! For this one, include the methodology assigned)
     + The name(s) of your teammate(s)
     + Summary at the top
       - Include a summary of your results and any significant findings or learnings
     + Honor pledge
   * Detailed results:
     + In this case, evidence that you’ve gotten all the tools up and working.
     + Test results so far for your programs.
     + Any other relevant results (like a list of test cases or the requirements you wrote)