Assessment Date: ___Spring 2013__________

Faculty Name(s): _____David Topham_______

1. Course Name and Number:
   CS102 - Introduction to Computer Programming Using C++

2. All Course SLOs from the Course Outline of Record
   1. Employ the basic syntax and semantics to write programs.
   2. Manage program flow by using conditional and iterative structures.
   3. Construct programs modularly from subroutines/functions.
   4. Construct programs with 'arrays', pointers, and other structures.
   5. Demonstrate basic stream and file Input/Output.
   6. Apply software development methodology to programming projects.

3. Specific Course SLO(s) assessed as part of this project:
   Apply software development methodology to programming projects.

4. Assessment strategy or tool used in the assessment.
   We asked students to answer the question below as part of our final exams over several semesters to determine if they were able to perform this basic application of software development:

Consider a grade-averaging scheme in which the final average of a student’s scores is computed differently from the traditional average if the scores have “improved.” Scores have improved if each score is greater than or equal to the previous score. The final average of the scores is computed as follows.

A student has n scores indexed from 0 to n-1. If the scores have improved, only those scores with indexes greater than or equal to n/2 are averaged. If the scores have not improved, all the scores are averaged.

The following table shows several lists of scores and how they would be averaged using the scheme described above.  Student Scores Improved?  Final Average

<table>
<thead>
<tr>
<th>Score List</th>
<th>Improved</th>
<th>Final Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>50, 50, 20, 80, 53</td>
<td>No</td>
<td>(50 + 50 + 20 + 80 + 53) / 5.0 = 50.6</td>
</tr>
<tr>
<td>20, 50, 50, 53, 80</td>
<td>Yes</td>
<td>(50 + 53 + 80) / 3.0 = 61.0</td>
</tr>
<tr>
<td>20, 50, 50, 80</td>
<td>Yes</td>
<td>(50 + 80) / 2.0 = 65.0</td>
</tr>
</tbody>
</table>

Given an array named 'scores', already populated with a number of scores (length), create pseudocode for the function hasImproved:

(a). finalAverage: function uses (b) and (c) to return either average scheme as described above
(b). average: function returns the average of the values in scores given a starting ('first') and an ending ('last') index.
5. Specific aspects of the assessment tool which link up to specific Course SLOs being assessed

SLO #6 is concerned the ability for a student to think at the design level. This is critical since a program cannot be written until the process is understand at a logic level. Independent of the language, pseudocode is the preferred way to solve a problem for a computer without being concerned with the particular language. This problem required that they had learned how to read a problem description and create a solution from it.

6. Results and analysis of the data. (Explain below and if applicable copy/paste any related documents, i.e. spreadsheets with data at the end of this document):

<table>
<thead>
<tr>
<th>CS102</th>
<th>No Clue</th>
<th>Some Proficiency</th>
<th>Close But No Cigar</th>
<th>Got It</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>14</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>3</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

7. Describe any faculty dialogue that occurred as part of the assessment process

The CS faculty met and discussed these results to determine if our current approach is successful at giving students the skills needed to design programs.

8. Next steps

We were not satisfied that this is the best that can be achieved by our students. We are especially concerned that students are not able to work out solutions based on self-study alone, and that they could benefit from more interaction with faculty and their peers to learn methods that work. We have changed our labs as an experiment to see if we get better results. Labs used to be self directed with some help from tutors, but now they are scheduled with instructor guidance to work through problems together. In addition, we have added team projects so students can learn from each other as well.

9. Results of implemented changes, if available at this time:

We plan to assess the skills again during spring 2014 to see if the new methods have increased the success rate.