Educational Program Institutional Effectiveness Summary 2008 - 2011

Degree Program: Bachelor of Science Computer Science
College/School: College of Arts and Sciences  Department: Computer Science
Submission Date: January 31, 2012

The BS Computer Science Option 1 received ABET accreditation in August 2011. The accreditation action extends retroactively from October 1, 2009.

Student Learning Outcomes
Students will demonstrate
1. knowledge of mathematical, algorithmic, and computing principles appropriate to the design and implementation of computer-based systems.
2. an ability to apply analysis, design, development, and testing principles in the construction of software systems.
3. an ability to function effectively on a software development team.
4. an ability to communicate effectively on technical subject matters.
5. a recognition of the value of continued professional development, as evidenced by the ability to acquire and use appropriate techniques, skills, and tools necessary for computing practice.
6. an understanding of professional, ethical, legal, and security issues and responsibilities, and the societal impact of computing.

Assessment Methods
1. ETS Major Field test administered in CSC 434: SLO 1
2. CSC 450 artifacts, including presentations, and rubric: SLOs 1, 2, 3, 4, 5, 6
3. Internship Supervisor Survey: SLO 3, 5
4. CSC 495 presentations and rubric: SLO 4
5. Alumni Surveys: SLO 5
6. Directed Individual Study presentations: SLO 5
7. CSC 385 artifacts: SLO 6

Summary of Assessment Results
SLO 1: Review of CSC 450 projects showed that they exhibited an understanding of the mathematical and computing principles that were appropriate to the design and implementation of common computer-based systems. This SLO is also assessed through the growing evidence from the ETS Major Field Test, which is taken by students in CSC 434 each fall. From this assessment, it appears that UNCW CSC students are on a par with institutions nationally. Specifically, the national institutional means (n=232) in the assessment categories Programming, Discrete Math and Algorithms, and Systems are 54.5, 34.7 and 41.7, respectively. The corresponding standard deviations are 13.6, 8.9, and 10.0. By comparison, UNCW CSC student averages were 58, 31, and 43, respectively. Thus, UNCW student are above the national average in Programming and Systems and within a third of one standard deviation of the average in Discrete Math and Algorithms.

SLO 2: The examination student projects in CSC 450 from 2008-2009 revealed that student code was not well documented. Student projects in CSC 450 have been assessed with respect to rubrics for Oral Communication, Written Communication, and Coding and Documentation. Performances with respect to these measurements for three academic years (2008-2009, 2009-2010, and 2010-2011) are summarized and provided in the documentation below. The greatest improvements appear in item 12 (Written - Graphics) and 13 (Written - Mechanics) which, in turn raised the overall average (item 14) in scores on the Written Communication Rubric. By contrast, there was a fall-off in performance observed with respect to item 21 (Coding – Efficiency). In addition, performance
measurements on item 19 (Coding – Reusability) and item 20 (Coding – Documentation) are below the threshold of acceptability and these results will be shared with the faculty on the Assessment Committee’s regular reporting schedule in August 2012.

SLO 3: The review of CSC 450 artifacts shows that students are capable of working on large multi-person projects; however the workload is often inequitable. Reading of the internship superior responses reveals that student interns function effectively on software development teams.

SLO 4: The review of CSC 450 artifacts revealed that projects showed that written code documentation was often missing or inadequate and that the written project reports contained stylistic and grammatical errors.

SLO 5: Internship supervisor surveys reveal that student interns are willing and capable of acquiring new skills as appropriate or necessary to the job, but do not provide insight into students’ ability to independently acquire and use new computing techniques.

SLO 6: CSC 450 projects demonstrated only minimal concern for internationally accepted software quality standards such as ISO 9126. During the 2010-2011 academic year, CSC 385 was revised and reorganized from a 1-hour course to a 3-hour course, and the assessment of SLO 6 was realigned with measurements of CSC 385. The first offering of the 3-hour course was in Fall 2011. While we have collected writing samples, those samples are scheduled for review later this semester.

**Implemented Improvements Based on Assessment Results**

Because the summary of results from the ETS Major Field Test (SLO 1) does not provide sufficient granularity of content knowledge and in order to improve overall student learning, the department faculty designed Faculty Course Reports (FCRs) and secured commitment to use FCRs to assess student learning outcomes in every course offering with the BS Computer Science Option 1 program.

In order to improve student software development processes and communication of technical material (SLOs 2 and 4), CSC 221 requirements were enhanced to require students to use Javadoc comments in all programming projects, and the assessment committee has recommended that Javadoc use become standard practice for all post-CSC 121 programming courses. In addition, CSC 385 has been revised to provide more opportunities for writing and oral presentations (SLO 4).

Based on the evidence for SLO 3 that workload within CSC 450 group project teams was often inequitable, the Assessment Committee recommended that instructors adopt practices to ensure more equitable distribution of workload and limit the group size to no more than three students. All but one of the projects during the 2010-2011 academic year had three students per team.

Based on SLO 6 evidence and Assessment Committee recommendations, ISO 9126 has been incorporated in to the competencies for CSC 450.

**Documentation of Implemented Improvements (below)**

- Summary of results for SLO 2
- Sample Faculty Course Report
- CSC 221 syllabus and course assignment showing inclusion of Javadocs
- CSC 385 syllabus showing changes resulting from the conversion from a 1-hour course to a 3-hours course, with writing assignments highlighted
- CSC 450 group project instructions with evidence of group size limits
- CSC 450 syllabus with evidence of addition of ISO 9126 standards being included
Results of SLO 2 2008-2009, 2009-2010, and 2010-2011

Comparison of Measurements

The sub areas of measurement are identified in the following table:

<table>
<thead>
<tr>
<th>Table. Rubric Area for Comparisons Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oral Communication Rubric</td>
</tr>
<tr>
<td>1 Oral - Organization</td>
</tr>
<tr>
<td>2 Oral - Subject Knowledge</td>
</tr>
<tr>
<td>3 Oral - Graphics</td>
</tr>
<tr>
<td>4 Oral - Mechanics</td>
</tr>
<tr>
<td>5 Oral - Eye Contact</td>
</tr>
<tr>
<td>6 Oral - Elocution</td>
</tr>
<tr>
<td>7 Oral Average</td>
</tr>
<tr>
<td>8 Written Communication Rubric</td>
</tr>
<tr>
<td>9 Written - Organization</td>
</tr>
<tr>
<td>10 Written - Professional Issues</td>
</tr>
<tr>
<td>11 Subject</td>
</tr>
<tr>
<td>12 Written - Graphics</td>
</tr>
<tr>
<td>13 Written - Mechanics</td>
</tr>
<tr>
<td>14 Written Average</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>
Sample FCR

Faculty Course Report for CSC 415 - Fall 2010

Gene Tagliarini

Student Learning Outcomes

The Student Learning Outcomes (SLOs) for CSC 415 are:

1. Students develop knowledge of representational issues and their relationship to applications of artificial intelligence.
2. Students learn and implement search methods, including depth-first, breadth-first, and heuristic search techniques, to find solutions for computationally intractable problems.
3. Students learn to model natural processes that perform computation.
4. Students implement computational paradigms that mimic natural processes.
5. Students develop knowledge of algorithm and implementation alternatives that enables them to choose appropriately.
6. Students develop skills in writing technical reports that describe findings that arise from application of software that they develop.

Course Assessment Schedule

Complete course-level assessment of CSC 415 will be conducted over a three-offering cycle. Course artifacts are gathered for each SLO as described in Table A.1 (Appendix A). As indicated in Table 1, Student Learning Outcomes 1, 3, and 5 were assessed during Offering 1 (Fall 2010) by using selected items from the final examination and a scoring rubric (Appendix B). During the second offering, SLOs 2, 4, and 6 will be assessed using a rubric for written projects. During Offering 3, all SLOs will be assessed by applying the written project rubric to the course research project.

<table>
<thead>
<tr>
<th>Offering</th>
<th>SLOs Assessed</th>
<th>Instrument(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Fall 2010)</td>
<td>1, 3, 5</td>
<td>Selected examination questions and rubric</td>
</tr>
<tr>
<td>2</td>
<td>2, 4, 6</td>
<td>Selected programming laboratory reports (e.g., Markov, k-means, nearest neighbor, Bayesian classifier, GA, or PSO)</td>
</tr>
<tr>
<td>3</td>
<td>1-6</td>
<td>Course Research Project</td>
</tr>
</tbody>
</table>
**Performance indicators**

A copy of the Final Examination itself appears in Appendix C and its associated rubric appears in Appendix B. **For each SLO, if at least 80% of the students score 3.0 or above, the SLO will be considered to have been met, and no further action is necessary.** Otherwise, the faculty member will review the offering, course materials, and assessment instruments with respect to each SLO for which the criteria have not been met. The review may lead to actions such as:

1. Specific recommendations for altering the course offering;
2. Collection of additional evidentiary measurements to clarify the degree to which the SLO is being met; or
3. Departmental discussion regarding the evidence, possible revision(s) of the course offering, and formulation of the SLO.

**Student Performance Data**

A copy of the final examination appears in the appendix. The results of applying the rubric to the final examination appear in Table 2.

---

### Table 2. Results of Applying Rubric to Selected Final Examination Questions

<table>
<thead>
<tr>
<th>Student</th>
<th>SLO 1</th>
<th>SLO 3</th>
<th>SLO 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>% of students scoring 3 or higher</td>
<td>83%</td>
<td>83%</td>
<td>83%</td>
</tr>
</tbody>
</table>

---

**Analysis and Response(s)**

Since the class performance with respect to each measured SLO meets the criteria for satisfactory performance, **no proposed course modifications are recommended at this time.** However, an unexpected side-effect of the measurement process did raise an item of interest for further investigation.
Students were allowed to elect whether to answer a question regarding genetic algorithms (GAs) or one pertaining to particle swarm optimization (PSO). The GAs question appeared on the test immediately before the PSO question. In addition, the GA question expected more of descriptive essay while the PSO question was focused more on execution of the PSO algorithm. Curiously, all students opted to answer the GA question rather than the PSO question. Consequently, during the next offering, I will plan to:

1. Reverse the order of presentation of GA and PSO questions, in order to determine if the response pattern arises strictly from location on the test;
2. Include in course-level assessment, programming lab exercise P6, which requires students to understand, implement, and experiment using the PSO algorithm.
# Appendix A - Mapping of SLOs to Course Artifacts

## Table A.1. CSC 415 Mapping of SLOs to Course Artifacts

<table>
<thead>
<tr>
<th>SLO</th>
<th>P1 Markov</th>
<th>P2 Nearest Neighbor</th>
<th>P3 K-Means</th>
<th>P4 Bayesian Classifier</th>
<th>P5 GAs</th>
<th>P6 Particle Swarm</th>
<th>Mid-term Test</th>
<th>Final Exam</th>
<th>Course Research Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Appendix B - Rubric

The 2010 Final Examination contains four parts, one for each of the following: historical context, a particle swarm optimization (PSO), a genetic algorithm (GA), and backpropagation (BP) training of a feedforward neural network (FFNN). This semester, the exam allows the student to choose alternatively between answering a question regarding PSOs or GAs. Thus, the rubric is subdivided for PSOs and GAs. The students’ performance on the PSO or GA and BP/FFNN sections will be used for course-level evaluation of SLO 1, SLO 3 and SLO 5.

<table>
<thead>
<tr>
<th>SLO</th>
<th>Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Students develop knowledge of representational issues and their relationship to applications of artificial intelligence.</strong></td>
<td>Student is unable to describe a GA for the test problem.</td>
</tr>
<tr>
<td></td>
<td>Student describes a representation for solutions, a fitness function, and progenitor selection.</td>
</tr>
<tr>
<td></td>
<td>In addition, the student describes population creation, crossover and recombination, and mutation operations</td>
</tr>
<tr>
<td></td>
<td>Student is unable to describe PSO algorithm design goals.</td>
</tr>
<tr>
<td></td>
<td>Student describes design goals.</td>
</tr>
<tr>
<td></td>
<td>In addition, the student calculates new velocity vector.</td>
</tr>
<tr>
<td></td>
<td>In addition, the student calculates complete particle update.</td>
</tr>
<tr>
<td><strong>2. Students learn and implement search methods...</strong></td>
<td>This SLO is not measured by the Final Exam.</td>
</tr>
<tr>
<td><strong>3. Students learn to model natural processes that perform computation.</strong></td>
<td>Student is unable to complete any steps in the BP algorithm</td>
</tr>
<tr>
<td></td>
<td>Student completes only the FF steps in the BP algorithm</td>
</tr>
<tr>
<td></td>
<td>Student completes FF and error signal steps in the algorithm</td>
</tr>
<tr>
<td></td>
<td>Student completes all steps in the BP algorithm and finds TSSE and RMSE</td>
</tr>
<tr>
<td><strong>4. Students implement computational paradigms...</strong></td>
<td>This SLO is not measured by the Final Exam.</td>
</tr>
<tr>
<td></td>
<td>Students develop knowledge of algorithm and implementation alternatives that enables them to choose appropriately.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>Student does not describe implementation alternatives for a PSO.</td>
</tr>
<tr>
<td>6.</td>
<td>Students develop skills in writing technical reports...</td>
</tr>
</tbody>
</table>
Appendix C—Measurement Instruments
I. Identify contributions of the following:
(Identify = describe in 2 to 4 sentences)

1. David Goldberg

2. Turing

3. Von Neumann

4. Kennedy and Eberhart

5. Werbos, Rumelhart, McClelland, Hinton, Le Cun, and Parker (What is their collective contribution?)
II. Problems

All students are required to complete the Backpropagation Problem. Undergraduate students may choose either the Genetic Algorithm or the Particle Swarm problem but not both. Graduate students are expected to respond to both the Genetic Algorithm and the Particle Swarm problems.

Genetic Algorithm (GA) Problem

Provide a complete and detailed description of a GA for the travelling salesman problem. Be sure to include:

1. The choice of representation for solution candidates.
2. The method for generating the initial population.
3. The crossover and recombination method.
4. A precise statement of the fitness function.
5. Other relevant decisions necessary for implementing a GA to find solutions.
6. **Particle Swarms**

1. What three behaviors are central to modeling swarm intelligence?

2. Suppose that some swarm contains bugs that can be represented by octets of binary digits (possibly bytes) and the following are currently true:
   - The state of the current global best solution is 1011 1101;
   - The state of the current local best solution for bug 43 is 1001 1100;
   - The state of the current solution stored in bug 43 is 1001 1100;
   - The constants $V_{\text{max}} = 3$, $\phi_1 = 0.8$, and $\phi_2 = 1.2$;
   - The current velocity vector $v$ for bug 43 is $v = (-4, -1, 3, 2, -1, -2, -3, -4)$;
   - The random value $\rho_{43,d} = 0.1*d$ for all dimensions $d \in \{1, 2, ..., 8\}$ of bug 43 (I know that this is unusual, but use these values for your test computations).

   1. What is the new velocity vector for bug 43?

   2. What is the next state of bug 43?
Backpropagation basics

Suppose that a 2-2-1 network is being used to learn a function using the three training patterns in the set \{((0.1, 0.2), 0.1), ((0.5, 0.4), 0.7), ((0.9, 0.3), 0.6)\}. Suppose also that the neurons use the sigmoid activation function \( f(x) = 1/(1 + e^{-\lambda x}) \), whose derivative \( f'(x) = f(x)*(1-f(x)) \), when \( \lambda = 1 \), which you may assume. Also assume the weights from the inputs to the hidden layer neurons and from the hidden to output layers are given by the matrices \( W_{hi} \) and \( W_{oh} \) respectively. Initially, the matrices \( W_{hi} \) and \( W_{oh} \) contain the weights \[ W_{hi} = \begin{bmatrix} -0.2 & 0.3 \\ 0.8 & -0.1 \end{bmatrix} \] (note that \( w_{12} = 0.3 \) is the weight from input 2 to hidden layer neuron 1) and \[ W_{oh} = \begin{bmatrix} 0.6 & -0.7 \end{bmatrix} \]. Suppose further that the biases are all initially set to 1.0, the bias weights are all set to 0.3, the learning rate \( \eta = 0.25 \), and that the third training pattern is being used for the first update sequence in the first epoch.

a. Draw the network and label the edges with the appropriate weights.

b. Calculate the output of each neuron

\( o_{h1} = \) ?

\( o_{h2} = \) ?

\( o_{o1} = \) ?

c. Calculate the error signal for the output neuron.

\( \delta_{o1} = \) ?
d. Calculate the error signal for each hidden layer neuron.

\[ \delta_{h1} = \text{_______________} \]?

\[ \delta_{h2} = \text{_______________} \]?

e. Calculate the weight adjustments for the connections from hidden neuron 1 to output layer neuron 1, from input 2 to hidden layer neuron 2, and the bias weight for hidden layer neuron 1.

\[ \Delta W_{o1,h1} = \text{_______________} \]?

\[ \Delta W_{h2,in2} = \text{_______________} \]?

\[ \Delta W_{h2,b} = \text{_______________} \]?

f. Assume that at epoch 1000 the training patterns were presented in the order given above and that the actual outputs for patterns p1, p2, and p3 were 0.15, 0.6, and 0.55, respectively. What are the TSSE and the RMSE?

TSSE = \text{_______________}?

RMSE = \text{_______________}?
## CSC 221 Syllabus and assignment indicating inclusion of Javadocs

### Spring 2012 CSC 221 course schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Week</th>
<th>Topic(s)</th>
<th>Book Chapters</th>
<th>Programming Assignment</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/11</td>
<td>1</td>
<td>Welcome. Recalling CSC 121</td>
<td>1-5</td>
<td>Programming Assignment 0</td>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td>2</td>
<td>Classes, Objects, Methods, Constructors</td>
<td>6</td>
<td>Programming Assignment 1</td>
<td>1/16 is a holiday 1/18 is last day to drop (without a grade)</td>
</tr>
<tr>
<td>1/23</td>
<td>3</td>
<td>One dimensional arrays + File I/O – text</td>
<td>8/9</td>
<td>Programming Assignment 2</td>
<td></td>
</tr>
<tr>
<td>1/30</td>
<td>4</td>
<td>Two dimensional arrays and Array Lists + File I/O - text</td>
<td>8/9 &amp; 12</td>
<td>Programming Assignment 3</td>
<td></td>
</tr>
<tr>
<td>2/6</td>
<td>5</td>
<td>Graphical User Interfaces: Swing components and Event Handling</td>
<td>7</td>
<td>Programming Assignment 4</td>
<td></td>
</tr>
<tr>
<td>2/13</td>
<td>6</td>
<td>GUI continued. Using the debugger + javadoc utility</td>
<td>7</td>
<td>Programming Assignment 5</td>
<td></td>
</tr>
<tr>
<td>2/20</td>
<td>7</td>
<td>Recap + test review</td>
<td>Test 1 on Wednesday 2/22 (Closed Book) and Friday 2/24 (Closed Book)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2/27</td>
<td>8</td>
<td>Test 1 returned. Inheritance</td>
<td>11</td>
<td>Programming Assignment 6</td>
<td>2/28 is last day to withdraw with W</td>
</tr>
<tr>
<td>3/5</td>
<td>9</td>
<td>Inheritance</td>
<td>11</td>
<td>Programming Assignment 7</td>
<td></td>
</tr>
<tr>
<td>3/12</td>
<td></td>
<td></td>
<td>SPRING BREAK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Day</td>
<td>Lecture Topic</td>
<td>Assignment</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
<td>---------------------------------------------------</td>
<td>--------------</td>
<td>--------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>3/19</td>
<td>11</td>
<td>Polymorphism, abstract classes, interfaces</td>
<td>11</td>
<td>Programming Assignment 8</td>
<td></td>
</tr>
<tr>
<td>3/26</td>
<td>12</td>
<td>File I/O – binary, object</td>
<td>13</td>
<td>Programming Assignment 9</td>
<td></td>
</tr>
<tr>
<td>4/2</td>
<td>13</td>
<td>File I/O continued + Exceptions</td>
<td>12</td>
<td>No class on Friday, 4/6</td>
<td></td>
</tr>
<tr>
<td>4/9</td>
<td>14</td>
<td>Recap + test review</td>
<td></td>
<td>Test 2 on Wednesday 4/11 (Closed Book) and Friday 4/13 (Open Book)</td>
<td></td>
</tr>
<tr>
<td>4/16</td>
<td>15</td>
<td>Search algorithms</td>
<td>16</td>
<td>Programming Assignment 10</td>
<td></td>
</tr>
<tr>
<td>4/23</td>
<td>16</td>
<td>Sorting algorithms + Recursion</td>
<td>16 &amp; 15</td>
<td>Student Team Project Presentations on Friday 4/27</td>
<td></td>
</tr>
<tr>
<td>4/30</td>
<td>17</td>
<td>Last day of class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/7</td>
<td></td>
<td></td>
<td></td>
<td>Final Exam 11:30 – 2:30 in CI 2006</td>
<td></td>
</tr>
</tbody>
</table>
Objective

To develop a software package that will function as a tool for designing simple Graphical User Interfaces using Swing components. The tool must allow

- the user to **design user interfaces** that include **one instance of a JPanel, and contain multiple instances** of JButton, JLabel, JTextField, JTextArea objects, attached to the JPanel.
- **save a designed user interface** to a user specified file so that it can be edited later.
- **export Java source code corresponding to the designed user interface** to a file which, when compiled and executed, would yield the user interface designed using the tool.

User Interface

Your software will include, at a minimum, the following user interface elements:

1. **A menu bar** with the following options (and sub-options):
   a. **File** (New, Open, Save, Export, Exit)
   b. **Edit** (Select, Cut, Move)
   c. **GUI** (JPanel, JButton, JLabel, JTextField, JTextArea)

Desired functionality

The software will have the following functionality and be graded as described below:

1. **Javadoc documentation:**
   - All classes and methods that you write must be documented using Javadoc. You will be submitting the documentation in html format along with your code. **You will receive a grade of 0 or**
Your final project grade is the \textbf{PRODUCT} of this grade and the grade you receive on the rest of the project.

\begin{itemize}
  \item Each class, and each method in the class, must include the \texttt{@author} javadoc tag showing who wrote that code.
\end{itemize}

2. \textbf{(20 points)} Allow user to design Swing user interfaces that contain Swing components specified above. The user will specify location and dimensions of the components using mouse interaction as specified below.

\begin{itemize}
  \item A left mouse press marks the location of the component.
  \item The user then indicates the dimensions of the component by dragging the mouse. As the user drags the mouse, the current dimension of the component is shown as a red rectangle.
  \item Mouse release marks the end of the process.
  \item The red rectangle disappears and the designated GUI component appears on the design pane.
\end{itemize}

3. \textbf{(10 points)} Allow user to save GUI to a user specified file. Allow user to read, display, and modify a GUI previously saved in a file.

4. \textbf{(10 points)} Completely erase GUI currently under construction. Tie this action to the File-New menu option.

5. \textbf{(10 points)} Component selection: Allow user to select a component that appears in the GUI being designed. The user indicates the component to be selected by a right mouse click within (or on) the component to be selected. The software responds by drawing a red rectangle around the selected component.

6. \textbf{(10 points)} Allow user to cut (remove) selected component from the GUI.

Note that the user has to select a component before it can be cut.

7. \textbf{(10 points)} Allow user to move a component.

Note that the user has to select a component before it can be moved. The user moves a selected component by pressing the mouse on the component and dragging the mouse. The component moves with the mouse drag.
1. **(10 points)** Allow user to specify the following properties for a component: name, label, attached Java code. For example, in the context of a JButton in the GUI, the properties might be "calcButton", "Calc", "calcButton.addActionListener(this)". Note that the user has to select a component before its properties can be set. All objects created must have default, generic, values for properties when created. For instance, the JButtons may be named button1, button2 etc. and have labels "b1", "b2", etc.

2. **(20 points)** Export source code to a user-specified file. The code must compile without further editing by the user and, when compiled, must generate a GUI identical to the one designed by the user.

   Tie this action to the File-Export menu option. Here is an example of a Java source file created by the tool and a screenshot of the corresponding user interface designed using the tool.

3. **(10 points) Extra Credit - add-a-feature.** You will receive up to 10 extra points for a feature that you add to the tool. How many extra credit points you receive depends on the utility and slickness of your enhancement.

---

**Software Design**

1. **Download the jarfile SwingMan.jar** and add it as an external jar file to your project. It contains the following:
   - The **class** BasicGUITool and
   - The **interface** SwingThing

2. The BasicGUITool class provides some basic functionality of a GUI design tool. You will implement a class corresponding to your design tool that **extends the BasicGUITool class.**

3. You will implement your own classes, like MyJButton, MyJFrame, MyJLabel etc. that extend the corresponding Swing components JFrame, JButton, JLabel, JTextField, JTextArea. **Each of these classes MUST implement the SwingThing interface.**

4. You may not modify SwingThing.java or BasicGUITool.java. Here is the [javadoc documentation](#) for BasicGUITool and SwingThing.
**Team composition and due date**

- Teams will be composed of three members each. Each team must demonstrate its project in class on Friday, December 2, 2011. There is no opportunity for demonstrating the project at any other time. If you miss the demonstration opportunity, you will not receive any credit for the project.

---

**Project Grading**

- Credit will given for completing the tasks identified in the section above. Each task will be scored on a boolean basis. That is, your software either fully provides the desired functionality or it does not. No partial credit will be given for any feature. Thus, if your code works only on Thursday afternoons when the wind is blowing out of the Northwest, you will not get any credit for that part.
- You must upload your complete Eclipse project, including Javadoc documentation, on Blackboard no later than 1:00 pm on Monday, December 5, 2011.
CSC 385 Supplement Indicating Writing Assignments

CSC 385: Professional and Ethical Issues in Computer Science

Blackboard9

- Home
- Text
- Description
- Objectives
- Grading
- Class Format
- Schedule

Instructor Information

Laurie Patterson
2025 CIS Building
Department of Computer Science
University of North Carolina Wilmington

910.962.3906 (voice)
910.962.7457 (fax)
pattersonL@uncw.edu

Office Hours:
Monday/Wednesday 1200-0200
Tuesday/Thursday 0100-0200
and by appointment

Blackboard 9
select UNC Wilmington
use email username (without @uncw.edu)
and SeaNet user PIN (six digits)

Class Meeting Time:
Tuesday and Thursdays
0930-1045
CIS 1006
Text, Supplies, and References

Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing
Herman T. Tavani, Wiley, 2010
ISBN-10: 0470509503
REQUIRED

Writing for Computer Science, 2nd Ed.
Justin Zobel, Springer, 2004
ISBN-10: 9781852338022
REQUIRED

The Elements of Style, 4th Edition.
Strunk & White, Pearson, 1999
ISBN 0-205-30902-X.
Recommended

Course Description

CSC 385. Professional and Ethical Issues in Computer Science (3) Prerequisite: ENG 101 or equivalent and junior or senior standing in computer science. Ethical and professional issues arising from the impact of computer science and related technologies on society. Topics include ethical issues, obligations of professional practice, privacy and security, intellectual property, work and health issues, and the impact of emerging technologies. Students give both oral and written presentations and participate in the discussion of case studies.
Course Learning Objectives/Outcomes

At the conclusion of the course the student will be able to:

1. Identify ethical issues as they impact computer science and related disciplines. [WI2]
2. Differentiate between the main ethical theories and be able to use the ethical theories in evaluating the ethical issues impacting computer science and related disciplines. [WI5] [WI2]
3. Discuss ethical issues in writing, using appropriate reference to the established Code of Ethics of the professional society relevant to that student's field (ACM, IEEE, etc), and apply professional codes of ethics to analyze and resolve ethical questions. [WI5] [WI2]
4. Demonstrate the ability to write within the computer science discipline including writing one or more research papers that demonstrate the students grasp of ethical issues, display a clear understanding of how the ideas of other persons may be properly cited and used in written documents, and illustrate use of popular formats for presenting published papers in computer science. [WI3] [WI4]
5. Prepare and present information on a technical topic, in a professional manner.
6. Identify and locate appropriate sources of information to support decisions and written ideas. [WI1]
7. Analyze and evaluate arguments using rules of logic and be able to formulate effective arguments based on sound premises. [WI2] [WI4] [WI5]

Mapping Student Learning Outcomes to Course Content

<table>
<thead>
<tr>
<th>Course Learning Objectives/Outcomes</th>
<th>Class Project Written</th>
<th>Small Written Assign</th>
<th>Class Project Present</th>
<th>Topic Present</th>
<th>Resume</th>
<th>Midterm</th>
<th>Final</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

27
Relation to program outcome: This course teaches the principles of ethical analysis and how technology, law, and ethics interact in society, to help the graduate confront and deal with the ethical challenges that arise in professional practice.

This course also improves the graduate's ability to communicate effectively with a variety of audiences, both technical and non-technical. This course is used in part to assess CSC’s achievement of ABET General Criteria E, F, G, and H (An understanding of professional, ethical, and social responsibilities; An ability to communicate effectively with a range of audiences; An ability to analyze the impact of computing on individuals, organizations, and society, including ethical, legal, security, and global policy issues; Recognize the need for, and an ability to engage in, continuing professional development).

**Academic Integrity:**

University Policy on academic integrity will be followed for this course. Cheating will be taken very seriously, resulting in harsh penalties. Since the skills required in this class are also required in the next class, cheating in this class will seriously hamper your ability to pass the next class.

The University of North Carolina Wilmington is a community of high academic standards where academic integrity is valued. UNCW students are committed to honesty and truthfulness in academic inquiry and in the mastery of existing knowledge. This commitment begins when new students matriculate at UNCW, continues as they create work of the highest quality while part of the university community, and endures as a core value throughout their lives.

It is the responsibility of every faculty member, student, administrator and staff member of the university community to uphold and maintain the highest academic standards and integrity of the university. Any member of the university community who has reasonable grounds to believe that an infraction of the Honor Code has occurred has an obligation to report the alleged violation to the faculty member teaching the class who, in turn, must report the allegation to the Office of the Dean of Students. This obligation is a core value of the Honor Code, and must be fulfilled by each and every member of the university.

UNCW students affirm their adherence to the Honor Code by signing an Honor Pledge after enrolling at the university; thereafter, each student must conform to the Honor Code at all times. The absence of signing the pledge does not excuse students from their obligation to read, practice and be held accountable to the rules and spirit of the Honor Code. Students who observe or suspect an Honor Code violation are to notify the instructor in whose course the alleged infraction occurred.

Please follow the policies below:

Appropriate Collaboration
• Sharing class notes with another student.
• Discussing anything that was covered in class.
• Helping a fellow student locate a bug in his program, provided the following are true:
  o You have already completed your program.
  o You never type or dictate code for the student.
  o You help with minor details, not solving the programming problem for him/her.

Inappropriate Collaboration (this is a partial list of examples only)

• Showing another student (who has not completed the assignment) your code.
• Copying code from another student.
• Verbally providing other students with the solution to the program. (This would be along the lines of giving them the key to solving the problem when they need to think it through themselves.)
• Helping other students during a test or quiz.
• Doing another student’s work.

To succeed, do more of the programming challenges at the end of each chapter than are assigned!

**Grading**

Grades will be determined on the basis of students’ performance on the items shown in the table below.

<table>
<thead>
<tr>
<th>COURSE ITEM</th>
<th>GRADE PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small writing assignments (5 or 6)</td>
<td>20%</td>
</tr>
<tr>
<td>Formal Written Research Paper</td>
<td>20%</td>
</tr>
<tr>
<td>Formal Research Paper Presentation</td>
<td>10%</td>
</tr>
<tr>
<td>Presentation of an Ethical Issue</td>
<td>10%</td>
</tr>
<tr>
<td>Resume &amp; Interview</td>
<td>10%</td>
</tr>
<tr>
<td>Midterm exam</td>
<td>10%</td>
</tr>
<tr>
<td>Final Examination + class Participation</td>
<td>20%</td>
</tr>
</tbody>
</table>

A modified 10-point scale will be used to compute your course grade. If your course score falls just below a cutoff, the higher grade may be assigned solely at the discretion of the instructor.
Factors that affect this judgment are the distribution of grades, improvement during the semester, and attendance.

- A plus/minus system will be used to assign final grades using the following scale:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93-100</td>
</tr>
<tr>
<td>A-</td>
<td>90-92</td>
</tr>
<tr>
<td>B+</td>
<td>87-89</td>
</tr>
<tr>
<td>B</td>
<td>83-86</td>
</tr>
<tr>
<td>B-</td>
<td>80-82</td>
</tr>
<tr>
<td>C+</td>
<td>77-79</td>
</tr>
<tr>
<td>C</td>
<td>73-76</td>
</tr>
<tr>
<td>C-</td>
<td>70-72</td>
</tr>
<tr>
<td>D+</td>
<td>67-69</td>
</tr>
<tr>
<td>D</td>
<td>63-66</td>
</tr>
<tr>
<td>D-</td>
<td>60-62</td>
</tr>
<tr>
<td>F</td>
<td>&lt;60</td>
</tr>
</tbody>
</table>

Incomplete grades are given rarely and only in very specific situations.

1. First, the student must be passing.
2. Next, the student must be able to complete the work of the course entirely on his or her own.
3. Finally, the student must be prevented from completing the course by verified, unforeseen circumstances beyond the control of the student.

These conditions must be documented and verified by both instructors before an incomplete grade may be given.

**Assignments/Class Format**

**Assignments:**

- **Small writing assignments (about 5 or 6): 25% of grade.**
  - These are assigned approximately one week before they are due. Expected length is 1000 words (about 4 pages of double-spaced 12-point type with 1-inch margins). These assignments will present the student with one or more readings addressing some current issue relating to ethics in technology or the impact of technology on society. The student will describe, evaluate and react to it in some way; supporting, critiquing, etc. These assignments are designed to help students hone their writing and evaluative skills.

- **Formal Written Research Paper: 10% of grade.**
  - The student will design and conduct a simple research assignment— for example, comparison of algorithms, analysis of spam email received, evaluation of software development tools or languages, etc. The assignment must gather and analyze data in order to answer some question. (There is no requirement the question be deep or even particularly difficult, but must be a serious question of interest.) The student will write up results in a manner consistent with submission for publication; that is, the write-up must include an abstract, introduction and discussion of the problem, explanation of how data was gathered, a summary of results, and discussion of what those results mean. Bibliographic references are required. This is a formal written paper with an expected length of about 2500 words, or 10 internal pages, (i.e. not including title page, bibliography, or charts/graphics) and 8-15 references. Formatting instructions for this
paper will be provided by the instructor. Each student paper will be developed in an iterative manner that allows for instructor feedback and student updates as needed.

- **Formal Research Paper Presentation: 10 % of grade.**
  - The student will make a presentation to the class on their research findings. The presentation will be evaluated by the instructor and by all students in the class. It is expected that the research will be completed prior to the presentation of findings. *Formal attire is required for all presentations.*

- **Presentation of Ethical Issues: 10% of grade.**
  - Each student will make one or more class presentations on an ethical issue. Students are expected to read any relevant material in the course text and supplement the material in the course text by sourcing other material through their own research efforts. As with the Formal Research Paper Presentation, the presentation will be evaluated by the instructor and by all students in the class. The final grade, however, will be determined by the instructor.

- **Midterm exam: 10% of grade.**
- **Final exam: 20% of grade.**
- **Class participation: 10% of grade.**
  - 5% of the grade is based on attendance, 5% on participation in class discussion. Repeated tardiness or absence will have an effect on course grade.
  - Regular class attendance is mandatory.
  - Completion of assignments will entail reading the textbook and researching various topics germane to the course. A student may be required to lead class discussion based on a scheduled lecture topic. Expect to spend several hours researching and formulating each topic so you can lead a discussion group.

- **Résumé: 5% of grade.**
  - In this assignment, you will create a résumé and letter of application that are geared toward a particular job advertisement or career fair event. You will use actual education and work-related information about yourself to create useful and realistic documents. This assignment will either help you to create a résumé and letter from “the ground up,” or it will help you to substantially improve documents that you have previously prepared.

- **Instructional Strategies:**
  - Students are expected to have read course material before class. Class sessions will focus on discussion of readings and exploration of alternative views. Any changes to assignments or changes in deadlines will be posted on the course webpage, along with assigned readings or required Internet links.
  - Last-minute or "breaking-news" readings may be distributed in class. This course makes heavy use of class discussion. You will be responsible for knowing what was discussed in class. You should not assume that doing the reading and showing up to class on test day will be adequate.
  - This is a writing-intensive course. Students should plan on spending substantial time weekly on preparing and revising assignments for the course. Likewise, there is a comparatively large amount of reading to be done for this course. Students who are having problems with the course should see me after the first or second assignment, so we can develop a plan for successful completion of the course. Research takes time; writing takes time. Be sure to allocate enough!
  - Late work is accepted only in documented cases of illness or other circumstance outside the student’s control. Any other arrangements for late submissions must be made in
advance. Any reconsideration of a grade must be requested within 5 working days after the grade is assigned.

- Evaluation criteria:
  - No assignments are dropped.
  - The small writing assignments are intended as "practice writing" and will be graded relatively lightly; however, the formal research report and final paper are considered formal writing, and graded as such.
  - Draft submissions are used to provide feedback to students and are not graded. However, if the draft for the midterm or final are not turned in, the grade for the finished product will be penalized.
  - Grading rubrics for assignments are posted on the course web page.

- Work Submission:
  - Assignments will be uploaded electronically as described on the course web page. Acceptable file formats, document layout, etc., will be discussed in class.

- Academic Honesty:
  - All submissions are expected to be your original, individual work.
  - Departmental policy will be strictly followed in cases of suspected plagiarism or other academic dishonesty. More information is on the course web page.
  - Due Dates for assignments will be announced in class and on the course web page.

- Classroom Etiquette & Expectations
  - We expect students to be civil & respect their professors and classmates.
  - Be on time to class. If you are going to miss class tell the professor but be aware this does not excuse your absence.
  - Cell phones, pagers, etc. are not appropriate in the class room. Turn them off and put them away.
  - Side conversations are distracting to others. Don’t do it.
  - The classroom should have an environment of professional courtesy.
  - Intellectual discourse involves challenging ideas in a civil and respectful manner.
  - Respect the learning of others.

**Deadlines**
A due date will be specified for every assignment when it is distributed. Late assignments will not be accepted. If you have an emergency that interferes with your ability to attend class or to complete an assignment on time, let me know as soon as possible and as far in advance as possible. Do so before the due date has passed. Once the deadline has passed, I will likely be unable to offer you an extension.

Blackboard is used for submitting assignments. Please note that Blackboard will cut-off accepting submissions by the due date and time. No late submissions will be accepted, period.

**Special Needs**
If you have a disability and need reasonable accommodation in this course, you should inform the instructor of this fact in writing within the first week of class or as soon as possible. If you have not already done so, you must register the Office of Disability Services in DePaolo (formerly Westside) Hall (ext. 2-3746) and obtain a copy of your Accommodation Letter. You
should then arrange a meeting to make mutually agreeable arrangements based on the recommendations of the Accommodation Letter.

**Student Resources**

- UNCW Labs
  - CIS is open evenings
- WebCT /Blackboard
- THE INSTRUCTOR!!!
- ACM SIG Proceedings Templates
- Ethics Student Companion Site

The Writing Center
The University Learning Center
DePaolo (formerly Westside) Hall
first floor, office #1056
910.962.7857

All writers need readers, and the more readers you have while drafting a paper, the better your paper can become. The Writing Center provides experienced peer readers for all UNCW students as they develop and improve their writing skills. The Writing Center is not remedial, but is designed for all student writers who want to improve their papers. Consultations are led by faculty recommended peer writing tutors who are trained to help students identify areas to improve and develop specific revision plans. Visit our website to schedule an appointment: [http://www.uncw.edu/writing.htm](http://www.uncw.edu/writing.htm)

Students can also receive electronic response to their developing papers through our Online Writing & Learning (OWL) program. Visit our website for a variety of writing resources: handouts, guides, useful links, and the Online Consultation link: [http://www.uncw.edu/stuaff/uls/owl.htm](http://www.uncw.edu/stuaff/uls/owl.htm)

Writing Center Hours:

- Sunday: 2pm – 10pm
- Monday – Thursday: 9am – 10pm
- Friday: 9am – 5pm

**Tentative Schedule (the specific dates will change as the class roster becomes static)**

Blackboard is used for submitting assignments. Please note that Blackboard will cut-off accepting submissions by the due date and time. No late submissions will be accepted, _period_.

33
<table>
<thead>
<tr>
<th>WEEK/DAY</th>
<th>TOPIC</th>
<th>ADDITIONAL READING</th>
<th>PRESENTER</th>
<th>HW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day01</td>
<td></td>
<td></td>
<td>Instructor</td>
<td></td>
</tr>
<tr>
<td>Day02</td>
<td></td>
<td></td>
<td>Instructor</td>
<td>HW1</td>
</tr>
<tr>
<td>Day03</td>
<td></td>
<td></td>
<td>Instructor</td>
<td>HW2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Instructor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chapter 2</td>
<td></td>
<td>Instructor</td>
<td>HW3</td>
</tr>
<tr>
<td></td>
<td>Writing</td>
<td></td>
<td></td>
<td>IEEE format</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FTP</td>
</tr>
</tbody>
</table>
Chapter 5: Privacy

Protecting Patient Privacy in the Information Age by David B. Kendall* MacDonald

Introduction & background should be submitted for review.

4.

02/07/12

Robertson Lovette

Words to Cut from your resume

Resume Writing Tips

Resume Homework
<table>
<thead>
<tr>
<th>Day</th>
<th>Chapter</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/02</td>
<td>6: Computer &amp; Network Security</td>
<td>Garner Arnold</td>
</tr>
<tr>
<td>11/02</td>
<td>Midterm Exam</td>
<td></td>
</tr>
<tr>
<td>16/02</td>
<td>7: Cybercrime</td>
<td>Smith, R</td>
</tr>
<tr>
<td>13/02</td>
<td>8: Intellectual Property</td>
<td>Tran</td>
</tr>
<tr>
<td>14/02</td>
<td>8: Intellectual Property</td>
<td>Smith, B</td>
</tr>
<tr>
<td>15/02</td>
<td>9: Commerce</td>
<td>Allen Jensen</td>
</tr>
<tr>
<td>16/02</td>
<td>10: The Digital Divide and the Transformation of Work</td>
<td>Keller</td>
</tr>
<tr>
<td>18/02</td>
<td></td>
<td>Jordan</td>
</tr>
<tr>
<td>Date</td>
<td>Chapter</td>
<td>Instructor</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>03/15/12</td>
<td>Community</td>
<td>Hall</td>
</tr>
<tr>
<td>03/17/12</td>
<td>Community</td>
<td>Nikolov</td>
</tr>
<tr>
<td>03/20/12</td>
<td>Community</td>
<td>Brown</td>
</tr>
<tr>
<td>03/22/12</td>
<td>Community</td>
<td>Manker</td>
</tr>
<tr>
<td>03/27/12</td>
<td>Emerging Technologies</td>
<td>Brown</td>
</tr>
<tr>
<td>03/29/12</td>
<td>Emerging Technologies</td>
<td>Manker</td>
</tr>
<tr>
<td>04/03/12</td>
<td>Course Review and Summary</td>
<td>Brown</td>
</tr>
<tr>
<td>04/05/12</td>
<td>Each final presentation will be for 20 minutes followed by 5 minutes</td>
<td>Brown</td>
</tr>
<tr>
<td>Date</td>
<td>Day</td>
<td>Notes</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------</td>
</tr>
<tr>
<td>04/10/12</td>
<td>Day24</td>
<td></td>
</tr>
<tr>
<td>04/12/12</td>
<td>Day25</td>
<td></td>
</tr>
<tr>
<td>04/17/12</td>
<td>Day26</td>
<td></td>
</tr>
<tr>
<td>04/19/12</td>
<td>Day27</td>
<td></td>
</tr>
<tr>
<td>04/24/12</td>
<td>Day28</td>
<td></td>
</tr>
</tbody>
</table>
| 04/26/12 |        | SPOTS! and work on questions you may have left to do!!!
|          |        | Written Project Report Due
|          |        | State Holiday |
Final Exam:
Tuesday, May 8
0800-1100
CIS 1006

© Updated CSC 121 Spring

Website template by Arcsin
CSC 450 Supplement Indicating Group Project Requirements

CSC 450 – Group Activities

**Group Activities**

The main project for this course must be completed as a group assignment. Students are expected to work together to meet specified deadlines, and overall group objectives. The size of each group should vary from three to five (5) persons. The minimum prescribed roles in each group are as follows:

- Project Manager
- Chief Requirements Engineer
- Chief Designer
- Quality Assurance Engineer

*All members are expected to participate in all activities. In addition, the role will vary for each development phase.*

**Functions of each role**

**The Project Manager** will be responsible for coordinating the activities of the group and for liaison with the instructor to resolve issue affecting the group from time to time. The project manager should be: professional and responsible, a good organizer and an effective communicator. His/her functions include:

1. Coordinating group activities
2. Overseeing and managing activities
3. Allocating roles, activities and responsibilities
4. Coordinating and managing documentation activities

**The Requirements Engineer** leads the requirements effort.

**The Designer** leads the design effort.

**The Quality Assurance Engineer** oversees test case design, validation of requirements, design, etc.

**Approval of the Group**

When individuals have decided to form a group, the names and id#s of the group members along with their designated roles should be submitted for approval. **Once the members of a group have**
been approved any change must also be approved. In effect this means that group members are stuck with each other until the end of the project. If a member is delinquent then the other members will have a chance to rate the performance of each member at the end of each phase of the project. Members cannot be ‘hired and fired’ by the groups.

**How Projects are Chosen**

Projects may be assigned by the instructor. If a team is allowed to propose a project of their liking, the project must be approved by the instructor. Work on a proposed project should not commence before it is approved. If a project is rejected then a new proposal must be made or appropriate modifications could be made to the first project.

**Presentations & Reports**

Each group is expected to make at least three (3) presentations at the end of the analysis, design and testing phases. All group members are expected to participate in all presentations. Formal attire is required for presentations. All documents submitted should contain on the first or second page:

- The group number
- The names of group members and their student identification numbers.
- The title of the document.

A report is to be produced for each of the phases specified above. **Immediately following a presentation the report should be handed in. Late reports are not acceptable!** Work done in the phase will be graded based on the presentation and the report. Presentations must be conducted professionally, both in terms of dress and conduct. It is suggested that practice sessions be conducted by each group prior to the actual presentation. This will allow an opportunity for self assessment and for fine-tuning aspects of the presentation.

Effective reporting is essential in the development of almost all automated systems. Format of project documents, font style, print quality, etc. should be common to all group members. This will ensure that documents created by different team members can be merged with a high level of professionalism. All information for the presentations and the printed reports must be correct and complete. Information should not be duplicated.

**Information Storage**

Backup copies of all project information should be made and kept by each project team members.

**Managing Your Project**

Plan ahead by scheduling the activities to be completed and attaching deadlines to each. Planning should be a group activity. The project manager must identify tasks to be performed and delegate responsibilities to group members. All activities must be given a time deadline. Individuals are expected to be responsible and professional in carrying out all assignments. The key to project management is:
1. Plan all activities
2. Have regular meetings to access progress
3. Monitor tasks and take corrective action to ensure the completion of activities.

**Deliverables**

1. The Requirements Analysis Report in the prescribed format and at the specified time.
2. The Design phase Report in the prescribed format and at the specified time.
3. The Development & Testing Phase Report in the prescribed format and at the specified time. This should include source code and executables, as well as user manuals.

**Tracking & Control Mechanisms**

1. Each group is required to maintain a folder, of all activities and documents related to the group work. The folder should contain the following minimum information:
   
   - For each meeting of the group:
     - names of persons present at the meeting
     - the time the meeting began
     - the time the meeting ended
     - the time that each member arrived at the meeting
     - the time that each member left the meeting
     - the objects of the meeting
     - what was achieved at the meeting.

   - All documents used by the group e.g. questionnaires.
   - All documents relating to the project plan.
   - All other project related documents.

2. These documents will be inspected from time to time by the instructor.

Retrieved from: [http://people.uncw.edu/simmonsd/courses/Spring2012/csc450/info/groupActivities.htm](http://people.uncw.edu/simmonsd/courses/Spring2012/csc450/info/groupActivities.htm)
CSC 450 – Format of Reports


**General Rules Applicable to All reports**

1. Each Figure and Table must be properly captioned, e.g. “Figure 1: Requirements Class Model”.
2. Each figure must be accompanied by appropriate textual description that highlights the salient features of the figure.
3. Each Figure must be properly cited in some portion of the text, for example, “A requirements class model for the application is shown in Figure 1”.

**Group Formation & Project Selection Report Format**

4. Name of group.
5. Title of project
6. Names & ID numbers of members.
7. Assigned roles of members
8. Description of problem

**Requirements Engineering Report Format**

- Cover page: name of group or title of project (1%)
• Names of group members (1%)
• Table of Contents (1%)
• Introduction – (11%)
  ■ Description of problem / Purpose of report
  ■ Project Scope & Objectives
  ■ Success criteria
• Software Project Plan (40%)
  ■ A list of all resources that will be needed for the product and their availability
  ■ A work breakdown structure that describes tasks
  ■ An estimate of the cost of the software based on your WBS and:
    o Lines of code
    o Function points
    o COCOMO II model
  ■ An estimate of other resources required for the tasks
  ■ A project schedule that identifies when tasks will be done (charts should reflect WBS).
    ■ GANTT chart
    ■ CPM or PERT chart
  ■ A list of persons/teams and other resources allocated to tasks – a responsibility matrix
  ■ A risk plan that identifies risks and how they will be handled
  ■ Any performance Issues
  ■ Any management or Technical Constraints
  ■ Project monitoring and control mechanisms
• Requirements/Analysis Models (40%)
  ■ Major Software Functions
  ■ Activity diagrams
  ■ Use Case Diagrams (>=20% “fully dressed”, <= 80% brief or casual)
  ■ Requirements Class Models
  ■ Requirements/Data Dictionary
  ■ Any Limitations or Constraints (hardware, software, cost, time, etc.)
Non-functional Requirements

List and description of all tools used in requirements process

Description of Analysis Process (what was done, where, how, by whom, etc.) (5%)

Major Problems Encountered

Bibliography (1%)

Each team member must complete and submit a printed copy of the TEAM RUBRIC form for each team member directly to me (-5%).

Software Design Report Format

1. Cover page: name of group or title of project (1%)
2. Names of group members (1%)
3. Table of Contents (1%)
4. Introduction – (10%)
   o Purpose of this report
   o Update your introduction and problem description to include:
     a. A description of why the problem you are addressing is important.
     b. A description of similar projects done by others.
     c. The benefits of solving the problem that your project addresses.
     d. Include at least 10 references to published conference or journal articles.
     e. How the following issues will be addressed: Ethical issues, security, legal issues, societal impact.
     f. A sample introduction may be found here.
   o Project Scope & Objectives
   o Success criteria
   o Software Functions and sub-functions (r1, r1.1, r1.2, etc.) – a textual list of major software functions.
5. Background and Related Work (10%)
o **Background**: Identify areas which a novice would need to grasp before they could properly understand your project and write four or five pages of background description. Be sure to cite at least 8 references in addition to the two text books used for the course. Sample abstract, introduction and background may be found [here](#).

o **Related Work**: In this section you describe any published projects similar to yours and how your project differs, or if it does not differ, why you have undertaken your project. Sample related works sections may be found in many published articles.

6. **Updated Software Project Plan (10%)**

   a. Ensure that each diagram is accompanied by supporting text and that each diagram is cited in the text.

   b. Update the LOC, FP and COCOMO II estimates based on your design experience – it is very likely that the estimates may changes as your understanding of your project becomes clearer.

7. **Updated Requirements/Analysis Models (10%)**

   a. Update each diagram and description to ensure that each item reflects changes that have occurred as the project progressed from its inception to this point.

   b. Ensure that each diagram is accompanied by supporting text and that each diagram is cited in the text.

   c. Be sure to indicate how you plan to achieve the non-functional requirements identified in the requirements engineering phase.

7. **Design models (45%)**

   7.1. General Design constraints

   7.2. Architectural design using **at least two** architectural styles

      - Graphical model of architecture identifying subsystems.
      - Textual description of architecture, identifying strengths and weaknesses.
      - Subsystem design
- Textual description of subsystem identifying roles of subsystem and classes.
- Solution focused activity diagrams
- Design class diagrams
  - Pre and post conditions for each operation.
- Sequence diagrams
- State diagrams
- User interface design
- Other design documents / models
- List of design patterns used throughout the design.
- Justification for the architectural style to be used.
- Ensure that each diagram is accompanied by supporting text.
- If not yet done: describe how non-functional requirements will be achieved and how ethical, security, legal, societal will be addressed.

7.3. Updated Data dictionary
7.4. List and description of all tools used in requirements process
7.5. Description of design Process (what was done, where, how, by whom, etc.). Include a responsibility matrix.
7.6. Major Problems Encountered

8. Test Plan (10%)
   - General testing approach (e.g., top-down, black-box, etc.)
   - List of test cases along with expected results for the classes/methods.

9. Bibliography (2%)

10. Each team member must complete and submit a printed copy of the TEAM RUBRIC form for each team member directly to me (-5%).
**Software Implementation & Testing Report Format**

*(Written in a style similar to published articles – here are two Word templates: ACM LNCS You may use either.)*

1. Title of paper *(2%)*
2. Names of authors and their affiliations *(2%)*
3. Abstract *(2%)*
4. Updated Introduction, Background & Related Work – *(4%)*
   - Purpose of Report –
   - *If not already done* - Updated introduction from design phase to address:
     a. Ethical issues related to your software / How such issues are addressed / Security recommendations.
     b. Legal issues that relate to your software / How such issues are addressed / recommendations.
     c. Security issues that relate to your software / How such issues are addressed / Security recommendations.
     d. Societal impact of your software.
5. Updated Project Plan and Requirements Models from Design Phase *(10%)*
   - Update the LOC, FP and COCOMO II estimates. Include a table showing the original estimates during requirements engineering, the estimates during design, and the actual values now that the software is implemented. Provide comments as to the reasons for the differences and any lessons learned.
   - Updated Requirements/Analysis Models
     - Use Case Diagrams with all use cases “fully dressed”.
     - Ensure that each diagram is accompanied by supporting text and that each diagram is cited in the text.
     - Be sure to indicate how the non-functional requirements identified in the requirements and design phase were achieved.
Update each diagram and description to ensure that each item reflects changes that have occurred as the project progressed from its inception to this point.

6. Updated Design models *(20%)*
   
a) Ensure that each diagram is accompanied by supporting text and that each diagram is cited in the text.
   
b) Update each diagram and description to ensure that each item reflects changes that have occurred as the project progressed from its inception to this point.

7. Implementation *(55%)*
   
o List software functions implemented
   
o List functions not implemented with justification
   
o Show complete user interface design (menus and submenus)
     
     a. List options implemented
     
     b. List options not implemented with justification
   
o Updated Test Plan
     
     a. Actual testing approach used (e.g., top-down, black-box, etc.)
     
     b. **Substantial list of test cases** including expected and actual results and who performed the tests. Indicate which tests failed and whether or not the errors were corrected.
     
     c. List of other tests (including test cases) to be performed.
   
o Description of implementation & testing Process (what was done, where, how, by whom, etc.). Include a responsibility matrix.
   
o Major Problems Encountered
   
   o **User Manual**

8. Bibliography *(5%)*

9. **Each team member must complete and submit a printed copy of the TEAM RUBRIC form for each team member directly to me (-5%).**

10. **Use the written communication rubric and the coding and documentation rubric at the end of the document to help you prepare your code and written document.**
NB: Submit both hard soft copies of code, executables and report.

**PRESENTATION Format:** for Software Implementation & Testing

(Use the oral presentation rubric at the end of the document to help you prepare for the presentation.)

1. Opening page
2. Group name/members/roles
3. Problem description/project objectives
4. Success criteria
5. Presentation outline
6. Team member responsibilities
11. Requirements Models
   a) GANTT charts
   b) Use case diagram
   c) WBS
12. Design models
   a) Architectural models
   b) Design class diagram
   c) Examples of pre and post conditions of operations.
   d) State and sequence diagrams (give an overview, not all the diagrams)
13. Implementation
   o User interface
     a. List options implemented
     b. List options not implemented with justification
Test Plan

a. General testing approach (e.g., top-down, black-box, etc.)
b. List of test cases for the classes/methods tested, along with expected and actual results and who performed the tests.

Lessons learned – what do you take away from the course

14. Bibliography

15. Do a Demo

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

RUBRICS

<table>
<thead>
<tr>
<th>Oral Communication Rubric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oral - Organization</td>
</tr>
<tr>
<td>Oral - Subject Knowledge</td>
</tr>
<tr>
<td>Oral - Graphics</td>
</tr>
<tr>
<td>Oral - Mechanics</td>
</tr>
<tr>
<td>Oral- Eye Contact</td>
</tr>
<tr>
<td>Oral - Elocution</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Written Communication Rubric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Written - Organization</td>
</tr>
<tr>
<td>Written - Professional Issues</td>
</tr>
<tr>
<td>Subject</td>
</tr>
</tbody>
</table>
### Ethical Issues

### Legal Issues

### Security Issues

### Societal Impact

<table>
<thead>
<tr>
<th>Written - Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Written - Mechanics</td>
</tr>
</tbody>
</table>

### Coding and Documentation Rubric

*Editor's note: These are the ISO 9126 standards*

<table>
<thead>
<tr>
<th>Coding - Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding - Readability</td>
</tr>
<tr>
<td>Coding - Reusability</td>
</tr>
<tr>
<td>Coding - Documentation of code</td>
</tr>
<tr>
<td>Coding - Efficiency (Data Structures and Algorithms)</td>
</tr>
<tr>
<td>Coding - Usability</td>
</tr>
<tr>
<td>Coding - Reliability</td>
</tr>
<tr>
<td>Coding - Maintainability</td>
</tr>
<tr>
<td>Coding - Portability</td>
</tr>
</tbody>
</table>

Retrieved from:  