Appendix

Near-Final Draft of

Department of Computer Science

And Operations Research

Assessment Manual

Version: May 1, 2013

**Contents**

1. Service Course Assessment
2. Formative Assessments
3. B.A. and B.S. Program Outcomes
4. Performance Criteria
5. Old Curriculum: Table of Courses Satisfying Objectives
6. New Curriculum: Table of Courses Satisfying Objectives
7. Rating Rubrics for Performance Criteria
8. Graduate Degrees
   1. Program Objectives
   2. Mapping Objectives to Courses
   3. Rubrics for Graduate Assessment

Service Course Assessment

The NDSU General Education Committee has published seven objectives for General Education courses. These objectives are:

1. Communicate effectively in a variety of contexts and formats;
2. Locate and use information for making appropriate personal and professional decisions;
3. Comprehend the concepts and perspectives needed to function in national and international societies;
4. Comprehend intra=personal and interpersonal dynamics;
5. Comprehend concepts and methods of inquiry in science and technology, and their applications for society;
6. Integrate knowledge and ideas in a coherent and meaningful manner;
7. Comprehend the need for lifelong learning.

Here is a mapping of our General Education courses to these objectives:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course** | **Obj. 1** | **Obj. 2** | **Obj. 3** | **Obj. 4** | **Obj. 5** | **Obj. 6** | **Obj. 7** |
| CSci 114 |  |  |  |  | X | X |  |
| CSci 116 |  |  |  |  | X | X | X |
| CSci 122 |  |  |  |  | X | X |  |
| CSci 125 |  |  |  |  | X | X |  |
| CSci 155 |  |  |  |  | X | X |  |
| CSci 159 |  |  |  |  | X | X | X |

Under Outcome 5, our General Education courses concentrate on the following methods of inquiry from the General Education Committee rubric:

1. Setup a study experiment to answer a question;
2. Analyze data/evidence to answer a question or support/or not the hypothesis;
3. Use methods of inquiry to solve a problem.

For Applications for Society, we concentrate on:

2. Apply science/technology to a problem;

4. Use methods of inquiry to address a problem.

Under Outcome 6, we concentrate on this part of Integrate knowledge and ideas in a coherent manner:

1. Identify and organize information relevant to a question or issue.

For Integrate knowledge and ideas in a meaningful manner, we concentrate on:

1. Identify significant patters from information relevant to a question or issue;
2. Integrate information to gain new insights relevant to a question or issue;
3. Integrate perspectives and points of view to gain new insights relevant to a question or issue.

For Outcome 7, we concentrate on:

1. Identify and explain the need for lifelong learning;

2. Analyze the need for lifelong learning from the perspective of rapidly changing knowledge;

3. Examine the need for lifelong learning in order to understand how contemporary knowledge can be translated into new contexts, or how it may not be valid in the future;

4. Evaluate the perspective of rapidly changing knowledge to understand the need for lifelong learning.

For our assessment of our General Education courses, we select a subset of Objective aspects and examine samples of student work. We assess each sample of student work using the following rubric:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Score 0** | **Score 1** | **Score 2** | **Score 3** | **Score 4** | **Score 5** |
| The work shows no use of the Objective aspect | The objective aspect is used incorrectly | The objective aspect use is incomplete | The objective aspect use mimics uses taught | The work uses the objective aspect in their own way | The work shows significant insight into the objective |

We expect students to score 2 to 4 on our assessment. Over time, we hope to see fewer 2’s and more 4’s and 5’s. Several s/cores of 0 or 1 indicate that we might need to make significant changes to the course.

Formative Assessments

Computer Science Department

Version: February, 2013

All sections of all courses are expected to perform at least one formative assessment during each semester. This assessment is to assist you in adjusting your course to improve student learning. The assessment should not have any impact on student grades, but you might wish to give a small amount of extra credit to ensure that students take it seriously.

Most formative assessments should be done during class time. They should be limited to no more than five to ten minutes. Usually, you should discuss the results in summary form and what you intend to change as a result with your students at the next class meeting.

The Department suggests that you perform one formative assessment during approximately the fifth week of the semester and another during approximately the tenth week. However, if you notice student difficulties, you may wish to perform these assessments earlier in the semester.

The results are for you. You do not need to share them with anyone else, but you should pay attention to them. If done properly, these assessments should be valuable in helping you improve your teaching and in improving student learning. If you are teaching a service course, your supervisor should discuss your formative assessment results with you soon after the assessment is done.

One excellent source of techniques that can be used for formative assessment is <http://pages.uoregon.edu/tep/resources/newteach/fifty_cats.pdf> . Numbers 5, 6, and 7 are particularly recommended for your use.

B.A. and B.S. Degrees

Program Outcomes

Adopted from ABET

1. **Apply Knowledge**

An ability to apply knowledge of computing and mathematics appropriate to the discipline;

1. **Analyze Problems**

An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;

1. **Develop Solutions**

An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs;

1. **Work in Teams**

An ability to function effectively on teams to accomplish a common goal;

1. **Understand Issues**

An understanding of professional, ethical, legal, security, and social issues and responsibilities;

1. **Communicate Effectively**

An ability to communicate effectively with a range of audiences;

1. **Analyze Impact**

An ability to analyze the local and global impact of computing on individuals, organizations, and society;

1. **Continue Development**

Recognition of the need for, and an ability to engage in, continuing professional development;

1. **Use Current Methods and Tools**

An ability to use current techniques, skills, and tools necessary for computing practices;

1. **Understand Tradeoffs**

An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;

1. **Build Software Systems**

An ability to apply design and development principles in the construction of software systems of varying complexity.

Performance Criteria

(Activities to Exercise the Program Objectives)

December, 2008

|  |  |
| --- | --- |
| **Program Outcome** | **Performance Criterion** |
| 1. Apply Knowledge | 1. Solve problems using knowledge presented in the course; 2. Use course knowledge to solve test questions; 3. Improve a presented solution to handle situations more effectively; 4. Extend a presented solution to handle additional situations. |
| 1. Analyze Problems | 1. Practice Requirements elicitation; 2. Given a description of a problem, determine the requirements of a solution; 3. Explain why a given problem should or should not be solved with software; 4. Practice Requirements Analysis. |
| 1. Develop Solutions | 1. Given a problem, design and implement a solution; 2. Critique a given design for software; 3. Carefully test and debug a computer application; 4. Use metrics to evaluate a software implementation. |
| 1. Work in Teams | 1. Discuss a problem or question in small groups during a class period and prepare a report of conclusions; 2. Divide a software development project among a team of students and complete it; 3. Work as a class to solve a problem cooperatively; 4. Work in small teams to solve a set of problems during a week or longer; 5. Work in small teams to explore and evaluate alternative approaches to a situation. |
| 1. Understand Issues | 1. Discuss the issues involved in an ethical situation; 2. Prepare and present a short play or debate that explores the issues and viewpoints involved in an ethical situation; 3. Evaluate and improve a provided paper discussing the different viewpoints in a situation; 4. Prepare a paper justifying a specific position on an ethical dilemma. |
| 1. Communicate Effectively | 1. Give an effective Powerpoint or other presentation in a class; 2. Write a user manual for an application; 3. Write a test plan for an application; 4. Participate in a group discussion; 5. Prepare and conduct structured interviews to develop software requirements. |
| 1. Analyze Impact | 1. Discuss the impact of a category of software on its potential users and others; 2. Prepare a report on how a specific category of users are or will be effected by a particular type of computer use; 3. Prepare a report on how a specific type of software should be configured to best serve a specific community of potential users; 4. Evaluate which of two or more types of software would best serve society and more specific stakeholders. |
| 1. Continue Professional Development | 1. Answer a set of questions concerning the work described in a journal or professional publication; 2. Keep a list of professional publications and/or web sites consulted during the semester. Require the list to have a minimum size and diversity; 3. Give students a topic and some exercises to solve in that topic. Require students to learn about the topic on their own; 4. Require students to learn and use a software tool or language on their own; 5. Have each student prepare a plan of what they expect to learn (knowledge and skills) on their own during the first five years after they graduate and how. |
| 1. Use Current Methods and Tools | 1. Give students exercises using current tools; 2. Ask students to discuss or report on the advantages of a specific modern tool or method; 3. Ask students to demonstrate how to use a new tool or method to other students; 4. Have groups of students research and evaluate a new tool or method. |
| 1. Understand Trade-offs | 1. Have students develop multiple solutions to a problem and compare those solutions to each other using algorithm analysis; 2. Give students two or more solutions to a particular problem and ask them to justify their selection of one of those solutions using algorithm analysis; 3. Have students critique a given design or solution with respect to tradeoffs using appropriate theory and analysis; 4. Have students critique UML models using appropriate theory. |
| 1. Build Software Systems | 1. Either singly or in teams develop a software application using design and development principles; 2. Have students critique a software design using best principles; 3. Have students critique the source code of an application using best principles; 4. Have students improve a software design or source code using best principles. |

Old Curriculum

Table of Courses Satisfying Objectives

December, 2008 Version

The old curriculum was updated for students entering in fall, 2011. The old curriculum is still available for students who have not yet completed their programs and entered before fall, 2011. The old curriculum will no longer be available from fall, 2015 on.

Codes Used: B (baseline competency established); P (practice opportunities); E (exit competency evaluated).

Required Courses:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course | Oa | Ob | Oc | Od | Oe | Of | Og | Oh | Oi | Oj | Ok |
| 160 |  | B | B |  | B | B |  |  |  |  |  |
| 161 |  | P | P |  |  |  |  | B | B |  |  |
| 222 |  | P | P |  |  |  |  | P |  | P |  |
| 335 | P |  |  |  |  |  |  | P | P |  |  |
| 336 | P |  |  |  |  |  |  | P | P |  |  |
| 366 |  |  | P | B |  |  |  |  | P | B | P |
| 372 | B |  |  |  |  | P | B |  | P |  |  |
| 373 |  | P | P |  |  |  |  |  |  |  |  |
| 374 |  | P |  |  | P |  | P |  |  | P |  |
| 445 | E |  |  | E |  | E |  |  | E |  | E |
| 467 | P | P |  |  |  | P |  |  | P |  |  |
| 474 | P |  | P |  |  |  |  |  | P | E |  |
| 475 | P |  | E | P |  | P |  |  |  |  |  |
| 468 | P | E | E | P |  |  | P |  |  |  | P |
| 489 |  |  |  |  | E | E | E | E | E |  |  |

Elective Courses (3 must be taken):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course | Oa | Ob | Oc | Od | Oe | Of | Og | Oh | Oi | Oj | Ok |
| 413 |  | P | P | P |  | P |  |  | P |  |  |
| 418 | P | P |  |  |  |  | p |  |  | P |  |
| 426 | P | P | P |  | P |  |  | P |  |  |  |
| 453 |  | P |  |  |  |  |  | P |  | P | P |
| 454 |  | P |  |  |  |  |  | P | P |  | P |
| 458 | P |  | P |  |  | P |  |  | P |  |  |
| 459 |  |  |  | P |  |  |  | P |  | P | P |
| 469 |  |  |  |  | P |  | P | P | P |  |  |
| 476 |  |  |  | P |  | P | P |  |  | P |  |
| 477 |  |  |  |  | P | P |  |  | P |  | P |
| 479 |  |  | P |  | P | P |  | P |  |  |  |
| 488 | P |  | P | P |  | P |  |  |  |  | P |

New Curriculum

Table of Courses Satisfying Objectives

April, 2013 Version

This curriculum is required of students entering North Dakota State University in fall, 2011 or later. The old curriculum is still available for students who entered earlier, but will cease in fall, 2014.

Codes Used: B (baseline competency established); P (practice opportunities); E (exit competency evaluated).

Required Courses:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course | Oa | Ob | Oc | Od | Oe | Of | Og | Oh | Oi | Oj | Ok |
| 160 |  | B | B |  | B | B |  |  |  |  |  |
| 161 |  | P | P |  |  |  |  | B | B |  |  |
| 213 | B | P | P |  |  | P |  |  | P | B | B |
| 222 |  | P | P |  |  |  |  | P |  | P |  |
| 313 |  | P | P | B |  | P |  |  |  | P | P |
| 336 | P |  |  |  |  |  |  | P | P |  |  |
| 366 |  |  | P | P |  |  |  |  | P | P | P |
| 372 | P |  |  |  |  | P | B |  | P |  |  |
| 374 |  | P |  |  | P |  | P |  |  | P |  |
| 415 |  | P | P |  | P | P | P |  |  | P |  |
| 445 | E |  |  | E |  | E |  |  | E |  | E |
| 467 | P | P |  |  |  | P |  |  | P |  |  |
| 474 | P |  | P |  |  |  |  |  | P | E |  |
| 468 | P | E | E | P |  |  | P |  |  |  | P |
| 489 |  |  |  |  | E | E | E | E | E |  |  |

Elective Courses (3 must be taken):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course | Oa | Ob | Oc | Od | Oe | Of | Og | Oh | Oi | Oj | Ok |
| 371 | P |  | P | P |  |  |  | P |  | P | P |
| 413 |  | P | P | P |  | P |  |  | P |  |  |
| 418 | P | P |  |  |  |  | p |  |  | P |  |
| 426 | P | P | P |  | P |  |  | P |  |  |  |
| 436 | P | P | P |  | P |  | P | P | P | P |  |
| 453 |  | P |  |  |  |  |  | P |  | P | P |
| 454 |  | P |  |  |  |  |  | P | P |  | P |
| 458 | P |  | P |  |  | P |  |  | P |  |  |
| 459 |  |  |  | P |  |  |  | P |  | P | P |
| 469 |  |  |  |  | P |  | P | P | P |  |  |
| 473 | P | P | P | P | P | P | P | P |  |  |  |
| 476 |  |  |  | P |  | P | P |  |  | P |  |
| 477 |  |  |  |  | P | P |  |  | P |  | P |
| 479 |  |  | P |  | P | P |  | P |  |  |  |
| 488 | P |  | P | P |  | P |  |  |  |  | P |

Rating Rubrics

For Performance Criteria

February, 2009

The performance criteria are more fully described in another document.

If your class is large, assessing a randomly selected sample of ten student submissions is fine. All student submissions must be graded, however. The grades must be sufficiently important in the course grade to motivate students to do their best work.

In most cases, a 3 should indicate reasonable performance. A 5 or 0 should be unusual (less than 10% of scores). When a performance criterion is used to introduce a skill, however, scores of 0 to 3 should be expected with an occasional 4 or 5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Performance Criterion** | **0 Score** | **1 Score** | **2 Score** | **3** | **4** | **5** |
| 1. 1.Solve problem | Student does not seem to have understood the problem | Student understands the problem, but has made little progress towards a solution | Student has applied suitable methods, but has made significant errors | Student has applied suitable methods with minor mistakes | Student has successfully solved the problems using suitable methods. | Student has solved the problems and gone beyond the solution to describe its significance or limitations. |
| 2. Solve test question(s) | Student has addressed the wrong question(s) or no question at all. | Student understands what is being asked, but has made little progress towards a solution. | Student has started the solution correctly, but either has made significant errors or has not gotten very far. | Student has made minor mistakes or has not finished the solution. | Student has correctly answered the question completely. | Student has gone beyond the correct answer to place that answer in some relevant context. |
| 3. Improve solution | Student does not understand the problem and/or the presented solution | Student seems to understand the provided materials, but is unable to make progress on improvement | Student makes only minor progress on improvements. | Student makes significant progress on improvements, but is unable to complete. | Student completes the requested improvements. | Student makes the requested improvements and places the solution in some relevant context. |
| 4. Extend solution | Student does not understand presented materials | Student understands presented materials, but is unable to progress on extensions | Student makes only minor progress on requested extensions | Student makes substantial progress, but does not complete the extensions | Student completes the extensions. | Student completes the extensions and places the solution in a relevant context. |
|  |  |  |  |  |  |  |
| 1. 1. Elicitation | Student does not understand how to do elicitation | Student consults only himself or herself | Student consults only one or two potential users | Student uses only one elicitation method, but does a good job with that method | Student uses more than one method, but makes minor mistakes. | Student uses more than one method and gathers a good set of requirements |
| 2. Determine requirements | Student does not seem to understand the problem | Student makes significant mistakes | Student makes minor mistakes | Student determines the correct requirements, but does not organize them well. | Student determines correct requirements and organizes them well. | Student goes beyond determining requirements to determine some of their implications. |
| 3. Solve with software | Student does not seem to understand the problem. | Student makes only preliminary progress towards a solution. | Student makes significant progress towards a solution, but does not use best practices. | Student makes significant progress and uses best practices. | Student produces an effective solution. | Student produces an effective solution using best practices. |
| 4. Practice Analysis | Student does not seem to understand the problem. | Analysis is partial and trivial. | Analysis is incomplete, but shows substantial progress. | Analysis is effective, but poorly presented or does not use best practices. | Analysis is effective and uses best practices. | Analysis goes beyond what is expected to provide unexpected insight. |
|  |  |  |  |  |  |  |
| 1. 1. Design and implement | Student is unable to make any progress on design or implementation | Student makes minor progress on either the design or the implementation | Student makes significant progress on the design and the implementation. | Design and implementation are largely complete, but the design does not match the implementation | Design and implementation are complete and match. | Some analysis is done on either the complete design or the complete implementation. |
| 2. Critique design | Student misunderstand the design | Critique leaves most of the design unaddressed. | Critique addresses most of the design, but is superficial and/or inaccurate. | Good critique of the design with only minor errors or omissions. | Complete critique of the design according to best practices. | Complete critique together with useful suggestions for design improvements. |
| 3. Test and debug | Student is unable to get any tests to work. | Student does one test case. | Student does several test cases, but does not debug correctly. | Student does several test cases according to a reasonable methodology and debugs successfully. | Student adequately tests and debugs program according to best practices. | Student adequately tests and debugs program plus provides suggestions of development practices that could have reduced errors. |
| 4. Use metrics | Student does not seem to understand use of metrics. | Metrics not used appropriately. | Metrics used correctly, but no conclusions or wrong conclusions made. | Metrics used correctly and some correct conclusions made. | Metrics used correctly and appropriate conclusions drawn. | Metrics used correctly, appropriate conclusions drawn, and some analysis provided of why the calculated results occurred. |
|  |  |  |  |  |  |  |
| 1. 1. Discuss | Students do not work together. | Students work together, but do not accomplish very much. | Student team has accomplishments, but is dominated by one or two students. | Student team has full participation, but does not consider all the aspects expected. | Student team has full participation and completes the task assigned well, but the report is disorganized. | Student team has full participation and produces a well-organized report. |
| 2. Divide and complete | Students do not work together. | Students work together, but do not accomplish very much. | Student team has accomplishments primarily due to one or two members. | Team has full participation, but does not complete the project. | Team has full participation and completes the project successfully. | Team has full participation, completes the project successfully, and provides a useful analysis of their own teamwork. |
| 3. Work cooperatively | Students seem unable to cooperate. | Students work together, but do not accomplish very much. | One or two team members dominate the accomplishments. | Team has full participation, but does not solve the entire problem or makes undiscovered mistakes. | Team has full participation and solves the problem. | Team has full participation, solves the problem correctly, and provides useful analysis of their own teamwork. |
| 4.Work over time | Students do not work together. | Students work together, but do not accomplish very much. | One or two members dominate the team’s accomplishments. | Team has full participation, but does not complete all the problems correctly. | Team has full participation and completes all the problems correctly. | Team has full participation, completes all the problems correctly, and provides useful analysis of their own teamwork. |
| 5. Evaluate alternatives | Students do not work together. | Students work together, but accomplish little. | One or two members dominate. | Team has full participation, but does not effectively consider all alternatives. | Team has full participation and reasonably evaluates all alternatives. | Team completes assignment with full participation and goes beyond the assignment in some useful way. |
|  |  |  |  |  |  |  |
| 1. 1. Discuss | Student does not seem to understand the situation | Student can come up with only one issue for this situation and cannot discuss important ramifications of that issue. | Student can come up with one issue and discuss its implications. | Student can come up with more than one issue, but provides only superficial discussion. | Student can come up with more than one issue and discusses them well. | Student can come up with more than one issue, discuss them well, and provide a reasonable procedure for resolving the issues. |
| 2. Play or debate | Student(s) does not seem to understand the situation. | Student(s) uses only one issue. | Important issues are presented, but important implications are not. | Important issues and important implications are presented, but not fully explored. | Situation is fully explored. | Situation is fully explored and a reasonable procedure for resolving the issues is given. |
| 3. Evaluate and improve paper | Student does not seem to understand the important points of the paper at all. | Student understands many, but not all of the important points. | Student fully understands the paper. | Student understands the paper, but provides superficial or incorrect improvements. | Student provides useful improvements. | Student provides useful improvements and compares them to what is already in the paper. |
| 4. Prepare a paper | Student does not seem to understand the ethical dilemma. | Student understands the dilemma, but does not present a position. | Student presents a position with some support for that position. | Student presents a position and evaluates that position with respect to other positions, but is not convincing or the paper is poorly organized. | Student does an excellent job of presenting a position and evaluating it with respect to other positions. | Student presents a position and justifies it well. Student also generalizes the ethical dilemma and describes how their position and other positions would be evaluated in the more general dilemma. |
|  |  |  |  |  |  |  |
| 1. 1. Give presentation | Presentation is not on topic. | Presentation is not close to complete. | Presentation is complete, but poorly organized or presented. | Presentation is complete, well-organized and presented, but gives no analysis or evaluation. | Presentation is complete, well-organized and provides well-known analysis or evaluation. | Presentation is complete, well-organized, and provides innovative analysis or evaluation. |
| 2. Write user manual | Manual explains nothing. | Manual explains some use, but leaves out major functionality. | Manual is not  complete, but poorly organized or written. | Manual is complete, but poorly organized or written. | Manual is complete and well-written. | Manual is complete, well-written, and includes useful materials for a variety of potential users. |
| 3. Write test plan | Plan contains no tests. | Plan contains a few tests, but coverage is inconsistent and incomplete. | Plan contains a reasonable set of tests according to some methodology, but is poorly written or organized. | Plan contains a reasonable set of tests and is well-written and organized. | Plan organizes a reasonable set of tests according to priority based on some reasonable criterion. | Plan provides indications of the errors that can be found by each test. |
| 4. Group discussion | Group does not stay on topic for very long. | Group deals with the topic at a very superficial level. | Group has a reasonable discussion of the superficial aspects that does not have full participation. | Group goes beyond the superficial, but is dominated by only a few members. | Group has a good discussion with nearly full participation. | Group has a good discussion that leads to useful conclusions with full participation. |
| 5. Structured interviews | Student does not understand how to conduct an interview. | Only one or two interviews are conducted and they are not well-structured. | More than two interviews are conducted, but they are not well-structured. | More than two well-structured interviews are conducted, but the student does not formulate a reasonable set of requirements. | More than two well-structured interviews of a variety of potential stakeholders are conducted, but the resulting requirements are not effective or well-organized. | Nearly all important potential stakeholders are represented in the interviews and the resulting requirements are well-organized and effective. |
|  |  |  |  |  |  |  |
| 1. 1. Discuss impact | Student’s comments indicate a lack of understanding | Student does not contribute to the discussion, but does follow it. | Student’s contributions are limited to agreeing with previous speakers | Student makes small contributions | Student makes significant contributions to the discussion. | Student provides analysis showing significant insight. |
| 2. Prepare report | Student does not appear to understand important characteristics of the users and/or the software | Student describes the users and the software correctly, but not the impact. | Student misses many important impacts, but does correctly describe some impacts. | Student describes all the important impacts, but provides no analysis. | Student provides only minimal analysis. | Student provides significant analysis. |
| 3. Configuration report | Student does not understand the assignment. | Student describes the configuration features of the software, but not how it should be configured for this community. | Student describes how the software should be configured, but makes significant mistakes in this description. | Student correctly describes how the software should be configured, but does not explain why. | Student provides some justifications for how the software should be configured. | Student provides justifications which demonstrate significant insight and analysis. |
| 4. Evaluate alternatives | Report is limited to describing the alternatives, and makes mistakes doing so. | Report correctly describes the alternatives, but does not address the service to society or specific stakeholders. | Report asserts how each alternative would serve society and/or specific stakeholders, but does not justify the assertions. | Report provides incorrect or incomplete justifications. | Report provides adequate justifications and analysis. | Report provides excellent justifications and analysis. |
|  |  |  |  |  |  |  |
| 1. 1. Answer questions | Response shows lack of understanding of the paper and/or the questions. | Response addresses all questions, but merely quotes the paper. | Response shows some understanding of some aspects of the paper. | Response shows understanding of all important aspects of the paper. | Response includes explanation, examples, or justifications beyond those in the paper for several of the questions. | Response provides significant insight into the work reported in the paper which is not already contained in the paper. |
| 2. Keep list | List is not available. | List is too small. | List is not sufficiently diverse. | List meets minimum requirements for size and diversity. | List goes well beyond the minimum requirements for size and diversity. | List shows the student has searched for relevant other articles on at least some of the topics of articles on the list. |
| 3. Learn topic | Student has not explored the topic. | Student has done only one of the exercises assigned. | Student has done some, but not all of the exercises assigned. | Student has completed all the exercises assigned. | Student has gone beyond the exercises assigned to learn more. | Student has completed all assigned exercises and has a plan to continue learning in this topic. |
| 4. Learn tool or language | Student has not explored the tool or language. | Student cannot use the tool or language. | Student can use the tool or language, but not effectively. | Student can use the tool or language effectively. | Student understands when and how this tool or language should be used instead of other tools or languages they already knew. | Student understands why and how the limitations of this tool or language exist. |
| 5. Development Plan | Student does not have a development plan. | Student’s development plan is very vague. | Development plan has a good list of skills, but not a good idea of how they will learn those skills. | Development plan is good in some areas and not very good in others. | Development plan is very good. | Development plan includes why the student believes learning these skills will be important. |
|  |  |  |  |  |  |  |
| 1. 1. Use current tools | Student cannot or does not use the appropriate tools. | Tools are used ineffectively. | Tools are used inefficiently. | Tools are used effectively and efficiently, but not everywhere they should be used. | Tools are used well. | Student has developed ways to combine these tools which were not intended by the tool developers, but which are effective. |
| 2. Discuss or report on advantages | Report describes the tool, but does not list any advantages. | Report lists only one or two advantages. | Report only lists the advantages. It does not justify them. | Report lists advantages and their justifications from the Help file of the tool or tutorial on the method. | Report provides some advantages based on the student’s own experience and not from the Help or tutorial. | Report provides advantages based on comparisons with other tools or methods with overlapping purposes. These advantages are not just from the tool or method documentation. |
| 3. Demonstrate | Student is unable to demonstrate the tool. | Some parts of the demonstration do not use the tool or method correctly. | The demonstration is not well-organized. | The demonstration is well-organized, but incomplete. | The demonstration is well-organized and covers the major aspects of the tool or method. | The demonstration is well-organized and provides insight into why the tool or method works as it does. |
| 4. Group evaluation | This student did not contribute significantly to the evaluation. | The evaluation is incomplete and has significant errors. | The evaluation is correct, but incomplete. | The evaluation is correct, nearly complete, but not very well-organized. | The evaluation is well-organized, correct, and complete. | The evaluation provides insight on why the tool or method operates as it does and/or some implications of this operation. |
|  |  |  |  |  |  |  |
| 1. 1. Compare solutions | The student is unable to produce more than one solution. | The student produces some incorrect solutions to the problem. | The student’s solutions are correct, but there is little comparison. | The comparisons are asserted without justification. | The comparisons are justified. | The comparisons provide insight into the problem itself. |
| 2. Justify selection | Student does not make a selection. | Student makes a choice, but does not justify it. | The justification is incorrect. | The justification is correct, but misses significant aspects. | The justification is good. | The justification shows insight into the nature of the problem and/or the solutions. |
| 3. Critique design or solution | The student provides no critique. | The critique does not use tradeoffs correctly. | The critique does not use appropriate theory and analysis. | The critique uses appropriate theory and analysis, but incorrectly or incompletely. | The critique is accurate, complete, and uses appropriate theory and analysis correctly. | The critique justifies the use of the theory and/or analysis employed. |
| 4. Critique UML models | The student provides no critique. | The critique does not use theory. | The critique uses theory, but incorrectly or uses the wrong theory. | The critique uses proper theory correctly, but is incomplete. | The critique is accurate, complete, and uses proper theory. | The critique explains how the critique could be addressed to reduce or eliminate the problems with the diagram. |
| 1. 1. Develop using principles | The student or team is unable to complete the application. | The student or team completes the application, but does not document the design and/or development. | The student or team completes the application and documents, but does not use principles everywhere. | The team or student uses principles incorrectly in some places. | The team or student uses principles appropriately throughout. | The team or student justifies exceptions to principles and explains how the principles benefited the development effort. |
| 2. Critique using principles | The student does not do a critique. | The critique is incorrect. | The critique is not based on appropriate principles. | The critique uses principles incorrectly in some places. | The principles are used appropriately throughout. | The student gives some insight into the value of using these principles. |
| 3. Critique source code from principles | The student does not do a critique. | The critique is incorrect. | The critique is correct, but not based on best principles. | The critique uses best principles incorrectly in some places. | The best principles are used appropriately throughout. | The critique explains how the code could be improved to better satisfy the best principles. |
| 4. Improve using principles | The student does not make any improvements. | The improvements are not correct. | The improvements are correct, but not based on best principles. | The best principles are used incorrectly in some places. | The best principles are used appropriately throughout. | The student explains how the best principles guided the improvements. |

Program Objectives

Graduate Degrees

Version: February, 2013

1. Graduate Certificate in Software Engineering
   1. Students understand the present state of software development;
   2. Students understand the significant differences among informal projects and formal projects;
   3. Students can do each of the phases of software development;
   4. Students understand the advantages of an iterative development process compared to a sequential process.
2. Graduate Certificate in Electronic Commerce
   1. Students understand the impact of computers and computer networks on commerce;
   2. Students understand the present state of electronic commerce;
   3. Students can evaluate an electronic approach to commerce.
3. Master of Software Engineering (MSE)
   1. Students recognize good practices in software engineering;
   2. Students can read and understand the professional literature in software engineering;
   3. Students understand the present state of a significant area of software engineering;
   4. Students can develop useful applications using state of the art methods.
4. Master of Science in Computer Science
   1. Students understand the current status of a major area of computer science;
   2. Students can read and understand the academic literature in computer science;
   3. Students can evaluate specific research in computer science;
   4. Students can extend the state of the art in a specific area of computer science through their own research.
5. Master of Science in Software Engineering
   1. Students understand the present state of software engineering;
   2. Students can read and understand the academic literature in software engineering;
   3. Students can analyze specific research in software engineering;
   4. Students can extend the state of the art in a specific area of software engineering through their own research.
6. Ph.D. in Computer Science
   1. – d .Same first four as for M.S. in Computer Science;
7. Students can present and defend research in a specific area of computer science to their colleagues;
8. Ph.D. in Software Engineering
   1. – d. Same first four as for M.S. in Software Engineering;
9. Students can present and defend research in software engineering to their colleagues.

Mapping Objectives to Courses

Version: March, 2013

Codes Used: B (Baseline Competency established and some practice provided); P (Practice to build competency); E (Exit from program competency level)

1. Graduate Certificate in Software Engineering

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course** | **Oa** | **Ob** | **Oc** | **Od** |
| 713 | B | B | B | B |
| 714 | P | P |  | P |
| 715 | P | P | P |  |
| 716 | P | P | P | P |
| 717 | P | P |  | P |
| 718 | P | P | P | P |
| 790 |  | E |  | E |
| Project | E |  | E |  |

1. Graduate Certificate in Electronic Commerce

Has been inactive for several years.

1. Master of Software Engineering

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course** | **Oa** | **Ob** | **Oc** | **Od** |
| 713 | B | B | B | B |
| 714 | P |  | P |  |
| 715 | P |  | P | P |
| 716 | P |  | P | P |
| 717 | P |  | P | P |
| 718 | P |  | P | P |
| 724 |  | E | P | P |
| 746 | P | P | P |  |
| 747 | P | P | P |  |
| 765 |  | E | P | P |
| 793 (6 credits) | E |  | E | E |

1. Master of Science in Computer Science

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course** | **Oa** | **Ob** | **Oc** | **Od** |
| 713 | B | B | B |  |
| 724 | P |  | P |  |
| 741 | P | P | P | B |
| 765 |  | P | P |  |
| 790 |  | P | P |  |
| Other Courses (vary by student) | P | P | P | P |
| Comprehensive | P | P | P |  |
| Final Examination on Thesis or Paper | E | E | E | E |

1. Master of Science in Software Engineering

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course** | **Oa** | **Ob** | **Oc** | **Od** |
| 713 | B | B | B |  |
| 716 | P | P | P |  |
| 715 or 718 | P | P | P |  |
| 765 |  | P | P |  |
| Other courses (vary by student) | P | P | P | P |
| 790 |  | P | P |  |
| Comprehensive | P | P | P |  |
| Final Examination on Thesis or Paper | E | E | E | E |

1. Ph.D. in Computer Science

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course** | **Oa** | **Ob** | **Oc** | **Od** | **Oe** |
| 713 | B | B | B |  |  |
| 724 |  | P | P |  |  |
| 741 | P | P | P |  |  |
| 765 |  | P | P |  |  |
| 790 | P | P | P |  |  |
| Other courses (vary by student) | P | P | P | P | P |
| Preliminary Examination | P | P | P | P | P |
| Qualifier Examination | P | P | P |  |  |
| Final Examination on Dissertation | E | E | E | E | E |

1. Ph.D. in Software Engineering

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. **Course** | **Oa** | **Ob** | **Oc** | **Od** | **Oe** |
| 713 | B | B | B |  |  |
| 716 | P | P | P |  |  |
| 715 or 718 | P | P | P |  |  |
| 765 |  | P | P |  |  |
| 790 | P | P | P |  |  |
| Other courses (vary by student) | P | P | P | P | P |
| Preliminary Examination | P | P | P | P | P |
| Qualifier Examination | P | P | P |  |  |
| Final Examination on Dissertation | E | E | E | E | E |

Rubrics for Graduate Assessment

Version: April. 2013

1. Course Oriented Assessment Rubrics

Baseline values are expected to be one or two, although there will be some higher scores, especially for students with previous undergraduate or graduate degrees or substantial professional experience. Practicing scores should be in the range 2 to 4 with rare, if any, 0’s and a few 5’s. Exit scores are expected to be 3 to 5 with a few lower scores when the student has been struggling.

Remember that which of these objectives are relevant for a particular student depends on which degree program that student is attempting. In any course, we are likely to have students from several degree programs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Performance Criterion** | **Score 0** | **Score 1** | **Score 2** | **Score 3** | **Score 4** | **Score 5** |
| **Objective:** Understand the present state | Does not associate development with a formal process | Confused about the current state | Misunderstand current development concerns | Can recite literature evaluations | Can explain current thinking in their own words | Can explain implications of current state |
| **Objective**: Differences among informal and formal projects | Thinks everything is informal | No idea when to do formal vs. informal | Can recite differences given by text and lectures | Can describe differences in their own words | Understands some implication of the differences | Can address some implications |
| **Objective**: Can do each phase of development | Does not understand the reason for phases | Skips or ignores some phases | Can recite the phases | Can follow instructions with guidance | Can follow instructions without guidance | Can innovate during some phases |
| **Objective:** Iterative versus sequential | Always sequential since it is simpler | Does not understand when each should be used | Can recite taught material | Can express in his or her own words | Understands the implication | Can address some implications |
| **Objective:** Recognize good practices | Does not understand what makes a practice good. | Cannot actually do the evaluation | Can list some good practices, but not use them | Uses good taught good practices | Can evaluate practices | Can improve a practice |
| **Objective:** Students understand the state of a CS area | Does not understand the question | Makes serious mistakes | Makes minor mistakes. | Can recite from taught material correctly | Can summarize relevant literature | Synthesizes his or her own answer |
| **Objective:** Develop useful applications | Can not develop applications | Can do part of development, but not all | Result has significant problems | Result is good, but the process is bad | Good result and process for familiar application areas | Successfully adapts to unfamiliar application areas. |

1. The Preliminary and Final Examinations

We expect the preliminary scores to range from 1 -3 in most cases. The final scores should range from 3 – 5 in nearly all cases. For the preliminary, a 2 should indicate reasonable performance. For the final, a 4 should indicate reasonable performance. Notice, that a student might pass the preliminary or the final examination without reaching reasonable performance on these assessment measures. Further, a student can fail the preliminary or final even when their assessment is reasonable or better. There are aspects to the preliminary and final which are not captured by these assessment measures.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Performance Criterion** | **Score 0** | **Score 1** | | **Score 2** | | **Score 3** | | **Score 4** | | **Score 5** |
| **Objective:** Students can present and defend research |  |  | |  | |  | |  | |  |
| Student presents in a well-organized manner | No organization is apparent | Items are presented after first needed | Some needed items are not presented | | Items not presented where needed | | No problems with organization | | Helps clarity | |
| Student explains the research clearly | Student does not understand | Cannot be understood | No examples | | Not enough examples | | Clear | | Superior | |
| Student can answer questions on the research | Student does not understand the question | Cannot answer | Answer is wrong | | Answer is incomplete | | Answer is good | | Answer provides insight beyond the work | |
| Student proposes ways to improve or extend the research | Fails to understand the need | Incorrect means | Incomplete means | | Student does not get the implications of extension | | Student understands the implications | | Student provides new insights | |
| **Objective**: Students can read and understand the academic literature |  |  |  | |  | |  | |  | |
| A student can understand what he or she reads | Student seems not to have read | Student misunderstands | Student can repeat the words | | Student can summarize in own words | | Student can relate this work to other work | | Student has insights not present in the work | |
| A student can use what he or she reads | Does not understand possible use | Can recite available uses | Can use in expected ways with some guidance | | Can use in expected ways without guidance | | Generalizes effectively | | Combines with other work for effective use | |
| A student can critique what they read | Does not understand | Can repeat critique in the work | Can critique with mistakes | | Can critique successfully with guidance | | Can critique without guidance | | Critiques and addresses critique | |
| A student can combine material from more than one source | Does not understand connections among works | Understands connections wrongly | Understands connections | | Can make the connections with guidance | | Can make connections without guidance | | Can form insights from connections | |
| **Objective**: Students can extend the state of the art in a specific area |  |  |  | |  | |  | |  | |
| Student can apply existing research | Does not understand | Understands, but cannot apply | Can apply with guidance | | Can apply without guidance | | Looks for applications | | Can characterize the domain | |
| Student can apply the existing research to a new problem | Does not understand where the research is applied | Understands application incorrectly | Can apply to a new problem with guidance in problem selection and application | | Can apply to a new problem with guidance in application | | Can select and apply without guidance | | Can modify the work to extend application | |
| Student can extend the research to address at least one of its limitations | Does not understand the limitations | Partly understands limitations | Understands implications of limitations | | Can overcome a limitation with substantial guidance | | Can overcome with little guidance | | Can overcome without guidance | |
| Student can develop and use a new approach | Does not understand the need | Understands the need, but not how to deal with it | Can do with substantial guidance | | Can do with minimal guidance | | Can do without guidance | | Can generalize the new approach or understand its limitations | |