Bloomsburg University

Computer Science Assessment Plan

2016

Program Mission Statement

The Department of Mathematics, Computer Science, and Statistics offers a Bachelor of Science degree in computer science. The curriculum is broadly based in core areas of computer science, with an emphasis on the design, analysis, and production of complex and reliable software. Graduates will be prepared to advance in computing careers and lead in technical endeavors, or to pursue an advanced degree in computer science.

Program Educational Objectives

Program educational objectives are broad statements that describe the career and professional accomplishments that the computer science program is preparing graduates to achieve.
Three to five years after graduation, our computer science alumni will be pursuing an advanced degree or they will:

1. be professionally employed in the computing field
2. communicate and collaborate effectively in a team environment
3. continue to grow professionally by adapting to new technologies and assuming leadership responsibilities

Periodic Review and Revision Process

The Computer Science Curriculum Committee will review our mission statement and program educational objectives once every five years. Input from constituents will be obtained and utilized during each review.

The primary source of constituents' input for this review will come from advisory board members during our annual advisory board meeting on career day. We will also look at alumni survey results and any comments received from employers not on our advisory board and the general public. Employers and the general public are welcome to email comments on any part of this document to the department chairperson (Currently Dr. Curt Jones, cjones@bloomu.edu).

Student Learning Outcomes

We have ten program learning outcomes listed under six categories.

- **Software Engineering**
    1. Students will demonstrate strong programming skills involving at least two object-–oriented languages.
    2. Students will be able to write a significant application that efficiently utilizes a database for data storage and retrieval.
    3. Students will be knowledgeable about software design processes and methodologies.

- **Operating Systems**
    4. Students will have a strong understanding of operating system concepts.

- **Hardware**
    5. Students will have a strong understanding of computer hardware concepts.

- **Problem Solving**
    6. Students will be able to determine what Abstract Data Type (ADT) should be used to solve a problem and what data structure should be used to efficiently implement an ADT.
    7. Students will be able to analyze the complexity of algorithms.
    8. Students will be able to solve programming problems.

- **Communication**
    9. Students will demonstrate oral and written communication skills necessary to read, write, and speak effectively about concepts in computing.

- **Ethics**
    10. Students will understand ethical and legal issues involving digital technology.

Computer Student Learning Outcomes Periodic Review and Revision Process

The computer science curriculum committee will review our Student Learning Outcomes once every five years. Input from constituents will be obtained and utilized during each review.

The primary source of constituents' input for this review will come from advisory board members during our annual advisory board meeting on career day. We will also look at the Graduating Senior Survey results, Alumni Survey results and any comments received from employers not on our advisory board and the general public. Employers and the general public are welcome to email comments on any part of this document to the department chairperson (currently Dr. Curt Jones, cjones@bloomu.edu).

Student Learning Outcomes Assessment Plan

Success in achieving the above outcomes is assessed through the administration of various direct and indirect measures including the Major Field Test in Computer Science, course embedded assessments, and exit surveys of graduating majors. Assessment results are regularly reviewed by subcommittees of the Computer Science Curriculum Committee. Performance results are reported to the Computer Science Curriculum Committee and appropriate curriculum changes are implemented.

Direct Assessment Summary

Learning Outcomes Summary:	**Direct Assessment Method Summary**
Software Engineering: (1) Students will demonstrate strong programming skills involving at least two object-–oriented languages. (2) Students will be able to write a significant application that efficiently utilizes a database for data retrieval and storage. (3) Students will be knowledgeable about software design processes and methodologies.	(1) Course embedded assessment based on a C++ programming project in COMPSCI 255 (C++ with Data Structures) – **C++ Assessment.** (1 & 2) Course embedded assessment based on a Java programming project in COMPSCI 221 (Advanced Java) – **Java Assessment.** (3) ETS Major Field Test in Computer Science (**MFTCS**).
Operating Systems: (4) Students will have a strong understand of operating system concepts.	**MFTCS** field test.
Hardware: (5) Students will have a strong understanding of computer hardware concepts.	**MFTCS** field test.
Problem Solving: (6) Students will be able to determine what Abstract Data Type (ADT) should be used to solve a problem and what data structure should be used to efficiently implement an ADT. (7) Students will be able to analyze the complexity of algorithms. (8) Students will be able to solve programming problems.	(6 & 7) Abstract Data Type (ADT) and Runtime Analysis Assessment. (8) Programming Problem Solving Assessment.
Communication: (9) Students will demonstrate oral and written communication skills necessary to read, write, and speak effectively about concepts in computing	Course embedded assessment will be administered with student project presentations and papers from our capstone course. (COMPSCI 480 -–-– Object-–Oriented Software Engineering). Standard rubrics will be utilized to assess both written and oral communication skills.
Ethics: (10) Students will understand ethical and legal issues involving digital technology.	Course embedded assessment will be administered in the Computer Ethics course. Students will be presented with a scenario related to the computer industry and be asked to conduct an ethical analysis. A committee from the department will use a standardized rubric to analyze solutions.

Indirect Assessment

The Computer Science program utilizes an **exit survey of graduating seniors** as an indirect assessment method. This survey covers all of our learning outcomes and allows the computer science program to determine our students' perception of their education at the time of graduation. We also have a **graduate survey** that is sent to students three years after graduation. This survey helps us determine if our former students continued their education and how they are advancing in their current positions.

Summary of All Assessment Methods

Assessment Method	Administered	Frequency*	Reviewed	Relevant Outcomes
Major Field Test in Computer Science Assessment	During course COMPSCI 480	Every spring semester	Following fall semester	1, 3, 4, 5, 6, 7, and 8
C++ Assessment	During course COMPSCI 255	Once every 3 years.	Following semester	1
Java Assessment	During course COMPSCI 221	Once every 3 years.	Following semester	1 and 2
ADT and Runtime Analysis Assessment	During course COMPSCI 355	Once every 3 years.	Following semester	6 and 7
Programming Problem Solving Assessment	During course COMPSCI 386	Once every 3 years.	Following semester	8
Communication Skills Assessment	During course COMPSCI 480	Once every 3 years.	Following semester	9
Ethics Assessment	During Course COMPSCI 360	Once every 3 years.	Following spring semester	10
Graduating Senior Survey Assessment	End of semester	Every Year	Following semester	All
Graduate Assessment	Online Survey	Once every 3 years.	Following semester	All
Employer Assessment	Online Survey	Once every 3 years.	Once a year	All
Advisory Board Meeting	Career Day	Once every 3 years.	After Meeting	All

*Please see the detailed assessment calendar on the next page

Computer Science Assessment Schedule

	Spring 2011	Fall 2011	Spring 2012	Fall 2012	Spring 2013	Fall 2013	Spring 2014	Fall 2014	Spring 2015	Fall 2015	Spring 2016	Fall 2016
Mission Statement and Program Educational Objectives review. Once every five years						C						
Student Learning Outcomes review. Once every five years.						C						
Major field Test in Computer Science Every spring.	C		C		C		C		C		C	
C++ Assessment. Once every 3 years.	C				C						C	
Java Assessment Once every 3 years		†				C						C
ADT and Runtime Analysis Assessment Once every 3 years.		C		C		C						
Programming Problem Solving Assessment Once every 3 years.	C		C		C		C				C	
Oral Presentation Assessment Once every 3 years.	C		C							C		
Written Assessment	C		C						C			

Once every 3 years.												
Ethics Assessment Once every 3 years.		†				C						
Senior Exit Survey Assessment Every spring	C		C		C		C		C		C	
Alumni Survey Once every 3 years.						C						P
Employer Survey Once every 3 years.						C						P
Advisory Board Meeting During Career Day					C	C						

C – Assessment activity completed ---- P – Planned ---- Results not yet available

† – Planned assessment activity not completed due to Bloomsburg University being closed for two weeks as a result of a major flood.

Assessment Details

I. Major Field Test in Computer Science

Our primary assessment tool is the Major Field Test in Computer Science (MFTCS) offered by ETS testing services (www.ets.org). This test is given to our graduating seniors every spring semester. It allows us to compare our students to students at other universities and gives us a valuable external measurement with objective scoring and norm--referenced data. The test covers a broad spectrum of computer science topics and allows us to see how we are doing with five of our learning outcomes. The MFTCS measures student proficiency in programming skills (learning outcome 1), software engineering (learning outcome 3), operating systems (learning outcome 4), hardware (learning outcome 5), discrete problem solving (learning outcome 6), and algorithms and data structures (learning outcome 7).

The test is required of students enrolled in our Software Engineering course. This was designed to help motivate students to take the test seriously. The computer science faculty review the information from the test at their first meeting of the fall semester. Dr. Jones and Dr. Coles are the faculty primarily responsible for this assessment tool.

II. C++ Assessment

This assessment is a locally developed tool that allows us to measure how well our students can design and create software solutions in C++. This project based assessment is tailored to the specific objectives we have for designing software solutions and supports our learning outcome for students to have programming skills in two different object--oriented languages (learning outcome 1). The project is given to students enrolled in COMPSCI 255. Students in this course are normally second semester sophomores. Completion of the project is a requirement of the course. The projects are evaluated by a subgroup of the computer science faculty who then present their results to the entire group. Dr. Coles, Dr. Khan, Dr. Lu, and Dr. Wynters are the faculty responsible for this assessment tool.

III. Java Assessment

This assessment is a locally developed tool that allows us to measure how well our students can design and create software solutions in Java. This project--based assessment is tailored to the specific objectives we have for designing software solutions utilizing a database for data storage and retrieval. This assessment tool also supports our learning outcomes for students to have programming skills in two different object--oriented languages and to be able to utilize a database in programming. The project is given to students enrolled in COMPSCI 221. Our assessments are timed so that the Java and C++ assessments are given to different groups of students. Students in COMPSCI 221 are normally first semester sophomores. Completion of the project is a requirement of the course. The projects are evaluated by a subgroup of the computer science faculty who then present their results to the entire group. Dr. Lu and Dr. Jones are the faculty responsible for this assessment tool.

IV. Abstract Data Type (ADT) and Runtime Analysis Assessment

We place selected questions on the final exam of our junior--level Algorithms and Data Structures course. The exact questions utilized are to be selected by the instructor of the course subject to approval by the Computer Science Curriculum Committee. We consider how many students get the problems correct and the overall average score of each student on each question. The instructor of Compsci 355, Analysis of Algorithms and Data Structures, is responsible for the administration of this assessment tool.

V Programming Problem-−Solving Assessment

We utilize a programming contest-−like assessment activity to assess the ability of our upper-−level majors to determine and properly sequence the basic logical steps needed to implement an algorithm. The word "contest" requires clarification: our students are not competing against one another, but the structure and administration of the event is similar to that of many high school and college programming contests.

Students are given five programming problems of increasing difficulty to solve individually in three hours. Problems are either correct or incorrect for this assessment activity; we do not consider partial credit. We analyze how many problems our students get correct to help determine their programming problem solving abilities. Dr. Coles is the faculty member responsible for this assessment activity.

VI. Communication Skills Assessment

This performance appraisal assessment tool is applied to student presentations at the end of our required course in software engineering. The software engineering course is normally taken during a student's last semester. Students are required to give a fifteen-−minute presentation at a department seminar and write a paper about their project experience. The results are reported to the entire group during the following fall semester. The instructor of Compsci 480, Object-−Oriented Software Engineering, is responsible for the administration of this assessment tool.

VII. Ethics Assessment

Our ethics assessment is performed once every three years in COMPSCI 360. Students are given typical software engineering scenarios and asked to identify the various individuals in the scenario and how well these individual understood and followed their professional code of ethics. The students write their solutions during our required ethics course. The instructor of Compsci 360, Computer Ethics, is responsible for the administration of this assessment tool.

VIII. Senior Exit Survey

Our graduating seniors complete a locally developed survey every spring semester. This survey allows the students to state their perception of how well the department satisfied its learning outcomes. The department chairperson summarizes the information and presents it to the Computer Science Curriculum Committee during the first meeting of the fall semester. The department office is responsible for this assessment tool.

IX. Alumni Survey

An email is sent to graduates by the department every three years. We target students who have graduated three to five years prior to our survey. This assessment tool allows us to see how our graduates are advancing in their careers and determine how many have furthered their education in graduate school. We also use this assessment to gauge the accuracy of our Computer Science Program Educational Objectives.

X. Employer Survey

The department office sends an email to employers identified by our Alumni Survey inviting them to complete our online survey. This assessment tool allows us to see how our graduates are advancing in their

careers. We also use this assessment to gauge the accuracy of our Computer Science Program Educational Objectives. We administer this survey once every three years.

XI. Advisory Board

The Computer Science Advisory board meets yearly in conjunction with our annual Career Day event.

Advisory Board Members along with their year of graduation, employer and current position

Len Kalechitz 2001, Solution Development Firm, LLC, Computer Scientist
Scott McCarty 1998, OPTiMO, Director of Information Technology
James Campbell 1998, Penn State University, Senior Unix Consultant
Matthew Quinn 2002, The Pennsylvania State University, Applied Research Laboratory
Mike Trelease 1998 and 2006, Geisinger Health System, Program Director
Dan Miller, 2005, Hershey Corporation, Application Development Director
Mitch Parker, 1997, Temple University Health System, Chief Information Security Officer

Appendix Assessment

Instruments and Rubrics

Department of Mathematics, Computer Science & Statistics

Oral Presentation Assessment

Speaker:_____ Evaluator: _____

Presentation Topic:_____ Date:_____

Evaluation Scale: 4 3 2 1

Exemplary Good Marginal Unsatisfactory

Presentation Style

Criteria	Score	Weight	Total
1. Personal appearance is appropriate.	_____	1	_____
2. Speaks clearly and with sufficient volume. Articulates words well.	_____	2	_____
3. Smooth transitions between topics. Limits the use of filler words ("ums").	_____	2	_____
4. Uses engaging vocalizations. Confident speaker.	_____	1	_____
5. Avoids distracting mannerisms. Did not rush the presentation.	_____	2	_____
6. Uses audience appropriate vocabulary, content, and style.	_____	2	_____
7. Maintains appropriate eye contact with audience.	_____	1	_____
8. Maintains audience interest.	_____	1	_____
Presentation Style Weighted Total			_____

Content

Criteria	Score	Weight	Total
9. Presentation includes introduction, body and conclusion.	_____	3	_____
10. Content is logically organized.	_____	3	_____
11. Visual aids or presentation materials enhance presentation.	_____	3	_____
12. Demonstrates subject knowledge, easily understands and answers questions on the topic. Clearly well prepared. Responds effectively to questions.	_____	4	_____
Content Score Weighted Total			_____

Weighted Total_____/100___

Computer Science Graduating Senior Survey
Bloomsburg University of Pennsylvania
Department of Mathematics, Computer Science, and Statistics

Note: Completion of this survey is required to complete your application to graduate. Information gathered from this survey will be used in the assessment of our Computer Science program.

Name			
Date			
Permanent Mailing Address			
Permanent Email Address			
1. I entered the computer science program as a:	New Freshmen ■	Transfer (From a Community college) ■	
	Transfer (From a 4-–year college) ■	Other ■	
2. How many semesters in our program did it take you to graduate? (If more than 8, please explain why)	Semesters	Explain:	
3. Did you participate in an internship? If so, describe.	Yes ■ No ■	Describe:	
4. Were you employed as an undergraduate? If so, where? And how many hours a week did you work?	Yes ■ No ■	Where: Hours a week:	
5. What sector are you headed for upon graduation?	Corporate ■	Consulting ■	Education ■
	Government ■	Graduate School ■	Other ■
6. Who will be your employer (Graduate School) upon graduation?			
7. What interval do you expect your salary to be in?	$0 -- $20,000 ■	$40,000 -- $60,000 ■	$80,000 -- $100,000 ■
	$20,000 -- $40,000 ■	$60,000 -- $80,000 ■	$100,000 or more ■
8. How do you feel our program has prepared you for your next step?	a. Very Prepared ■	b. Reasonably Prepared ■	
	c. Somewhat Prepared ■	d. Poorly Prepared ■	
9. If you feel inadequately prepared, tell us why.			

10. Describe what you liked least about our program?	
11. What did you like best about our program?	
12. What concrete suggestions do you have for the department to better serve our students?	

13. Please assess how well we have prepared you on the following criteria

• Object-–Oriented Programming Skills	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Programming skills in Java	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Programming skills in C++	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Ability to write a significant database application	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Knowledge of software design processes and methodologies	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Knowledge of operating systems concepts	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Understanding of computer hardware	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Problem Solving skills	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Proficiency in algorithms and data structures	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Proficiency in oral and written communication of technical information	Poor ■	Satisfactory ■	Good ■	Excellent ■
• Understanding of ethical issues related to computing	Poor ■	Satisfactory ■	Good ■	Excellent ■

14. Place any additional comments here.

The following is an exported survey from Qualtrics. It is formatted differently on the web when completed by our graduates.

Computer Science Program Alumni Survey

1 Your Name (Optional)

2 Your Email Address (Optional)

3 We would like to survey the supervisors of our graduates. If you are willing to have us ask your supervisor to complete a short survey, then please provide us with your supervisor's name and email address.

4 What was your year of graduation?

- 2015
- 2014
- 2013
- 2012
- 2011
- 2010
- 2009
- 2008
- 2007
- 2006)
- 2005
- Before 2005

5 What majors did you complete at Bloomsburg University? (check all that apply)

- Computer Science
- Digital Forensics
- Mathematics
- Other _____

6 What minors did you complete at Bloomsburg University? (check all that apply)

- Computer Science
- Digital Forensics
- Mathematics
- Statistics
- Other _____

7 What extra-–curricular activities did you complete at while at Bloomsburg University? (check all that apply)

- I completed an internship.
- I was involved with the ACM club.
- I completed an Independent Study course.
- I was involved in research with a faculty member.

8 We welcome any comments about your participation in extra-–curricular activities sponsored by the department. What was interesting? What was useful?

9 Which phrase best describes how well the CS major prepared you for your career?

❍❍ Very well prepared.

❍❍ Reasonably prepared.

❍❍ Somewhat prepared.

❍❍ Not very prepared.

❍❍ Not at all prepared.

10 How would you rate your abilities in the following areas?

	Excellent (4)	Good (3)	Satisfactory (2)	Poor (1)	N/A ()
Leadership Skills	❍❍	❍❍	❍❍	❍❍	❍❍
Ability to adapt to new technologies	❍❍	❍❍	❍❍	❍❍	❍❍
Ability to work in a team environment	❍❍	❍❍	❍❍	❍❍	❍❍
Object-– Oriented programming	❍❍	❍❍	❍❍	❍❍	❍❍
Java programming	❍❍	❍❍	❍❍	❍❍	❍❍
C++ programming	❍❍	❍❍	❍❍	❍❍	❍❍
Database design and implementation	❍❍	❍❍	❍❍	❍❍	❍❍
Software engineering	❍❍	❍❍	❍❍	❍❍	❍❍
Operating systems knowledge	❍❍	❍❍	❍❍	❍❍	❍❍
Computer hardware knowledge	❍❍	❍❍	❍❍	❍❍	❍❍
Algorithms and data structures knowledge	❍❍	❍❍	❍❍	❍❍	❍❍
Problem solving	❍❍	❍❍	❍❍	❍❍	❍❍
Oral communication	❍❍	❍❍	❍❍	❍❍	❍❍
Written communication	❍❍	❍❍	❍❍	❍❍	❍❍

11 Did you continue your education after graduating Bloomsburg University?

❑❑ I have not attended graduate school

❑❑ I currently attend or I have attended graduate school

❑❑ I earned a Masters Degree

❑❑ I earned or plan to earn a Doctorate Degree

12 We welcome any additional feedback you could provide on the Bloomsburg University Computer Science program.

The following is an exported survey from Qualtrics. It is formatted differently on the web when completed by our graduates.

Computer Science Program Employer Survey

1 Company Name (Optional)

2 Your Name and Position (Optional)

3 How many Bloomsburg University Computer Science Students do you supervise?

4 How would you rate Bloomsburg University graduates in the following areas?

	Excellent	Good	Satisfactory	Poor	N/A
Leadership Skills	00	00	00	00	00
Ability to adapt to new technologies	00	00	00	00	00
Ability to work in a team environment	00	00	00	00	00
Object-– Oriented programming skills	00	00	00	00	00
Java programming skills	00	00	00	00	00
C++ programming skills	00	00	00	00	00
Database design and implementation skills	00	00	00	00	00
Software engineering skills	00	00	00	00	00
Operating systems knowledge	00	00	00	00	00
Computer hardware knowledge	00	00	00	00	00
Algorithms and data structures knowledge	00	00	00	00	00
Problem solving skills	00	00	00	00	00
Oral communication skills	00	00	00	00	00
Written communication skills	00	00	00	00	00

5 We welcome any additional feedback you could provide on the Bloomsburg University Computer Science program or its graduates.

Student name: _____ **Evaluator Name: _____**

C++ Assessment Rubric

	UNSATISFACTORY 1	MARGINAL 2	GOOD 3	EXCELLENT 4	SCORE
Pointers, operations on linked data structures, memory management	There is little or no demonstrated understanding of how to perform dynamic memory allocation or manipulate pointers.	There are missing or grossly incorrect functions and/or obvious errors that could cause memory leaks.	There are subtle errors that could lead to memory leaks but all functions are implemented and functional.	There are no potential memory leaks. Destructor, copy constructor, and assignment operator implemented correctly.	
STL iterators and sorting algorithms	STL is not used.	An STL vector and indexing is used instead of the required list class.	An STL list and an iterator are used with at most minor errors.	An STL list and iterator are used correctly and the list of objects is sorted properly.	
File I/O	Does not read any information from the input file.	Does not use C++ stream objects for file I/O, crashes, and/or does not read and store all the data in the file.	Uses C++ stream objects for file I/O, successfully reads and stores all the data in the file.	Uses C++ stream objects for file I/O, successfully reads and stores all the data in the file, using the most appropriate kind of loop, and closes the file.	
Operator overloading (and complexity requirement for operator+)	There is little or no demonstrated understanding of how to overload operators and/or how to invoke them.	There are significant gaps in knowledge of how to overload operators and/or how to invoke them. Operator+ does not meet the complexity requirement.	The operator overloading is generally correct, but the complexity requirement for operator+ is not met.	The required operators are correctly overloaded, and the complexity requirement for operator+ is met.	
Templates	No attempt to implement a template class.	Major errors resulting in a non-functional template class, e.g., a member function is not a template function.	No major errors; the template class can be instantiated and is functional.	No functional errors, and uses recommended coding conventions.	
General OOP principles	Incorrect parameter and return value types, global variables or other details that subvert the idea of information hiding, incorrect use of const.	Highly non-cohesive interface. No understanding of when/why to declare references and methods const. Member functions not focused on their particular responsibilities.	Public interface contains one or two member functions not related to the concept represented by the class. Member functions or references not consistently declared const when they should be.	Parameters and return values are declared with appropriate types. Const is used where appropriate. No global variables or other hacks to violate information hiding. Clear separation of public interface and private implementation. Cohesive public interface.	
Clarity	There are significant deviations from coding standards throughout. Many parts of the code are undocumented, overly complex, and/or cannot be understood without judgment or guesswork.	There are significant deviations from coding standards. The code is disorganized or poorly documented, and difficult to understand in places.	The code is generally easy to read, but in some cases there may be insufficient documentation, unusual or inconsistent indentation, cluttered or overly complicated code, or other minor deviations from coding standards.	The code is professionally written: neatly organized, easy to read and understand, with correct indentation, reasonable choices for identifiers, and internal documentation to explain non-obvious details of the logic or its implementation.	

Student Name: _____ **Evaluator Name:** _____

Java Assessment Rubric

	UNSATISFACTORY 1	MARGINAL 2	GOOD 3	EXCELLENT 4	SCORE
Implementing Interfaces	No attempt to implement the *Comparable* interface	Incorrectly implemented the *Comparable* interface	The *Comparable* interface is implemented correctly in most instances and classes.	The *Comparable* interface is implemented correctly in all the appropriate classes.	
Object-Oriented Design	Difficult to follow design.	Some good design elements, but many design problems are evident.	Reasonable class design, but some design problems are evident.	Excellent class design throughout the entire project.	
Generic Class Design	No attempt to introduce generic types	Generic types are introduced, but there are many problems with their specifications and implementations.	Generic types are introduced and they are used correctly in most instances.	Generic types are introduced and the types are used correctly in all instances.	
Java Coding Style (Programs are available to check for coding style)	No style	Many style faults	Most style conventions are followed. Most identifier names are appropriate. Most constants declared correctly.	All coding follows standard style conventions. All identifier names are appropriate. All constants are declared correctly.	
JavaDoc Documentation	Minimal java documentation. Most methods are not completely commented.	Many methods are not correctly documented.	Most methods are commented correctly and completely.	Each method and class has appropriate descriptions. All meta tags are correctly completed.	
Code	Code does not execute.	Code executes, but many implemented methods do not perform correctly.	Most implemented methods perform correctly.	The entire program is correct. All methods are implemented correctly.	
Problem Solution	Many program requirements are not completed.	Most requirements are completed, but few are correct.	Solution is well done; most requirements are completed correctly.	All program requirements are completed. Program is easy to use.	

Student Name: _____ **Evaluator Name:** _____

ADT and Runtime Analysis Rubric

	UNSATISFACTORY 1	MARGINAL 2	SATISFACTORY 3	EXCELLENT 4	SCORE
Analysis of Iterative Algorithms	Less than 35% correct.	36 – 60% correct.	61-– 85% correct.	86 – 100 % correct.	
Analysis of Recursive Algorithms	Less than 35% correct.	36 – 60% correct.	61-– 85% correct.	86 – 100 % correct.	
Application of Critical Thinking to Choosing Appropriate ADTs, Data Structures, and Algorithms	Less than 35% correct.	36 – 60% correct.	61-– 85% correct.	86 – 100 % correct.	

Student Name: _____ **Evaluator Name:** _____

Writing Assessment Rubric

	UNSATISFACTORY 1	MARGINAL 2	GOOD 3	EXCELLENT 4	SCORE
Grammar and spelling	Many sentences have grammar or spelling errors.	Most paragraphs have a grammar or spelling error.	Most paragraphs have no grammar or spelling errors.	The entire piece has at most two grammar or spelling errors.	
Sentence structure	Run on and awkward sentences occur in most paragraphs.	Some run on and awkward sentences are present. Sentence structure varies little.	Very few run on and awkward sentences are present. Sentence structure is usually varied appropriately.	No run on or awkward sentences. Sentence structure is varied appropriately.	
Paragraph structure	Most paragraphs are incoherent.	Some paragraphs are structured appropriately.	Most paragraphs are structured and obviously coherent.	Every paragraph is begun, developed and concluded appropriately.	
Composition structure	Ideas appear haphazardly or incompletely or are not present. Relationships among ideas are not evident.	Ideas are present but often unrelated. Main points are not evident. Little flow through the composition exists.	Main points are evident and usually related in a logical fashion. Introduction and conclusion are present.	Subject is introduced. Main points are developed. Transitions are made. Conclusions follow from main points.	

Notes:
(1) Content must be graded separately.