

Bergen Community College

ASSESSMENT REPORT FORM FOR ACADEMIC PROGRAM

Assessment Period: Fall 2014 - Spring 2016

Department/Program: CSEIT / Computer Science

Department Chair: Prof. Emily Vandalovsky

Program Assessment Liaison: David Wang

Date Submitted: June 1, 2016

❖ **Program Description or mission/goal statement of the Department/Program:**

The Computer Science program offers an A.S. degree along with a certificate program. These programs provide fundamental knowledge in hardware, programming, database, system analysis and design, data structures and algorithms, and other computer science concepts.

The Computer Science Program strives to prepare the student for a transfer A.S. degree and/or a career in computer science. All programs provide a basic background in applications and concepts. Additionally, the program provides a solid foundation for future study towards a four-year degree in computer science, computer engineering, computer information systems, and/or other relevant fields.

❖ **Program Learning Goals/Outcomes:**

Upon successful completion of the A.S. Degree in Computer Science, the student will be able to:

Foundations

1. Understand the fundamental concepts of computer science, be able to work with them, and understand how they can be used in real-world applications.
2. Be able to name the major components of a computer system and explain what each does.
3. Know the criteria to use in evaluating a software package and be able to perform tasks utilizing current problem solving software packages.
4. Be able to explain fundamental networking concepts.

Programming

1. Be able to design a program using structured development techniques.
2. Know how to implement a structured design using functions.

3. Understand the structure and processing of single and multi-dimensional arrays.
4. Know the fundamental algorithms for creating and processing sequential access files and random access files.
5. Be able to organize and represent data using structures.
6. Understand the fundamental techniques for sorting and searching data sets.
7. Be able to design a program using object-oriented techniques.
8. Be able to implement classes and their data and methods;
9. Be able to write programs using encapsulation, inheritance, and polymorphism concepts;
10. Be able to implement graphical user interfaces using swings, multithreading, and exception handling;
11. Be able to implement applets in HTML files.
12. Be able to use data structures.

Database

1. Be able to describe the components of a database system and the relationships between them.
2. Understand the major models of data and the role that each plays in the development process.
3. Know how to do the conceptual design of a database using the Entity-Relationship model of data.
4. Be able to use normalization theory to determine the quality of the design of a relational database.
5. Know how to use structured query language to create, modify, and query a relational database.
6. Understand the various schemes for the physical organization of a database.
7. Know the functions and goals of database administration.

Data structures and algorithms

1. Understand the concept of an abstract data type.
2. Be able to select the appropriate data structure and design the corresponding operations to implement an abstract data type.
3. Know the fundamental order of magnitude growth rates and how they are used to measure the run-time efficiency of an algorithm.
4. Know the principles of pointers, dynamic memory management, and be able to apply these concepts in constructing dynamic data structures.
5. Know the fundamental properties of stacks and queues and be able to implement them using dynamic linked-lists.
6. Be able to incorporate recursive techniques in the representation and implementation of an abstract data type.
7. Know the fundamental properties of binary trees, binary search trees, and general trees and be able to implement them using dynamic data structures.
8. Be able to represent graphs and networks using adjacency matrices and adjacency lists, and implement them using the appropriate data structure.

9. Know the major sorting algorithms and be able to identify the advantages and disadvantages of each.
10. Understand the use of hashing techniques in the storage and retrieval of data.

Hardware/Computer organization

1. Know how to trace the content of the appropriate special purpose registers in the CPU as assembly language instructions go through the instruction execution cycle.
2. Understand basic logic circuits, and understand how full adders are used to perform binary addition.
3. Know the different memory management techniques used by the operating system and the advantages and disadvantages of each.
4. Understanding how character data and real numbers are internally represented, and the ability to represent signed integers two's complement and in packed format.
5. Be able to identify the different addressing formats and addressing modes discussed in class, and will be able to write short code segments that use these addressing modes.
6. Be able to identify the parts of an Assembly Language program.

System analysis and design

1. Know the stages and sub-stages in the systems development life cycle.
2. Be able to apply PERT/CPM techniques to project management, monitoring and control: Gantt Charts.
3. Understand the tools and techniques for conducting a preliminary investigation of a systems project.
4. Know the advantages and disadvantages of the various techniques for information requirements analysis.
5. Be able to construct a conceptual model of a system using data flow diagrams, data specification tools, and process specification tools.
6. Be able to construct a conceptual of a system database using the Entity-Relationship model of data.
7. Know how to convert a conceptual database model into a relational database model.
8. Understand the fundamental concepts of systems design.
9. Be able to apply transform analysis and use design criteria to convert a data flow diagram to a system structure chart.
10. Know the various module specification techniques.

SEMESTER 1: CREATING PROGRAM-LEVEL ASSESSMENT PLAN

1. Program Learning Goal(s) or Outcome(s) to be assessed (from the above section):

Know how to implement a structured design using functions in programming.

2. Means of Assessment:

Two Projects (**Appendix 1**) in CIS265 - Advanced Programming Concepts.

Feedback from Dean:

SEMESTER 2: DEVELOPING ASSESSMENT TOOL (s) and TIMELINE

3A. Describe or attach assessment tool (s), including sources of data, timeline for data collection and how data will be analyzed.

Will collect project#2 and project#3 from two sections of CIS265 - Advanced Programming Concepts offered in Spring 2015 by February, review/grade them by March, and analyze the data by April. Use a rubrics of comments (10%), prototype (10%), header (10%), and statements (70%) to grade each submitted project. Data will be aggregated to be analyzed and benchmarked with 3B.

3B. Desired results faculty would like to see.

70% of students who submit both projects are expected to have at least an average grade of 80% (see rubrics in 3A) from these two projects.

Feedback from CIE:

SEMESTER 3: COLLECTING AND ANALYZING DATA

4. Summary of Results (attach aggregated data table, survey tool, etc., to support the summary)

50% of students who submit both projects had at least an average grade of 80% (see rubrics in 3A) from these two projects (**Appendix 2**).

5. Recommendations for Improvement:

- a. Use more lecture time to cover structured design using functions.
- b. More exercises for students are to be assigned on this topic.

- **Feedback from Dean:**

SEMESTER 4: CLOSING THE LOOP AND SHARING KNOWLEDGE

6. Use of Results:

By doing #5, in this semester, 54% of students who submit both projects had at least an average grade of 80% (see rubrics in 3A) from these two projects (**Appendix 3**). Though the improvement is not big, steps in **5** will be taken in fall 2016 to hopefully and continuously increase the student success rate on this topic.

Feedback from CIE: Get input about rubric grading from all instructors teaching CIS265.

Appendix 1

Project#2 Winning Division

Write a program that determines which of a company's four divisions (Northeast, Southeast, Northwest, and Southwest) had the greatest sales for a quarter. It should include the following two functions, which are called by main.

- `double getSales()` is passed the name of a division. It asks the user for a division's quarterly sales figure, validates the input, then returns it. It should be called once for each division.
- `void findHighest()` is passed the four sales totals. It determines which is the largest and prints the name of the high grossing division, along with its sales figure.

Input Validation: Do not accept dollar amounts less than \$0.00.

Project#3 Overloaded Hospital

Write a program that computes and displays the charges for a patient's hospital stay. First, the program should ask if the patient was admitted as an in-patient or an out-patient. If the patient was an in-patient the following data should be entered:

- The number of days spent in the hospital
- The daily rate
- Charges for hospital services (lab tests, etc.)
- Hospital medication charges.

If the patient was an out-patient the following data should be entered:

- Charges for hospital services (lab tests, etc.)
- Hospital medication charges.

The program should use two overloaded functions to calculate the total charges. One of the functions should accept arguments for the in-patient data, while the other function accepts arguments for out-patient data. Both functions should return the total charges.

Input Validation: Do not accept negative numbers for any information.

Appendix 2

Spring 2015 - Two Sections of CIS265											
	Comm-10 Proj2	Prot-10 Proj2	Header-10 Proj2	Stat-70 Proj2	Comm-10 Proj3	Prot-10 Proj3	Header-10 Proj3	Stat-70 Proj3	AVG Score P1&2	80 Or Above	
stu1	10	10	10	70	10	10	10	70	100	Yes	
stu2	5	10	10	35	10	10	0	70	75	No	
stu3	5	5	5	35	5	5	5	35	50	No	
stu4	10	10	10	70	10	10	10	70	100	Yes	
stu5	10	10	10	70	10	10	10	70	100	Yes	
stu6	10	10	10	35	10	10	10	35	65	No	
stu7	10	10	10	70	10	10	10	70	100	Yes	
stu8	10	5	10	35	10	10	10	70	80	Yes	
stu9	10	10	10	70	10	10	5	70	97.5	Yes	
stu10	10	10	10	70	10	10	10	70	100	Yes	
stu11	5	5	5	70	5	5	5	35	67.5	No	
stu12	10	5	10	70	10	5	10	70	95	Yes	
stu13	5	5	5	35	10	10	10	35	57.5	No	
stu14	10	10	10	70	10	10	10	70	100	Yes	
stu15	0	0	0	0	0	0	0	0	0	No	
stu16	10	10	10	70	10	10	10	70	100	Yes	
stu17	10	10	0	70	10	10	10	35	77.5	No	
stu18	0	0	0	35	10	10	10	35	50	No	
stu19	10	10	10	70	10	10	10	70	100	Yes	
stu20	0	0	0	0	0	0	0	0	0	No	
stu21	0	0	0	70	0	0	10	70	75	No	
stu22	0	0	0	70	5	5	5	70	77.5	No	
stu23	0	0	0	0	0	0	0	0	0	No	
stu24	5	5	10	35	5	5	5	70	70	No	
stu25	10	5	10	70	10	10	5	70	95	Yes	
stu26	10	10	10	35	10	10	10	25	60	No	
stu27	5	5	5	35	5	5	5	35	50	No	
stu28	10	10	10	70	10	10	10	70	100	Yes	
stu29	5	5	10	0	10	5	5	35	37.5	No	
stu30	5	10	10	70	5	10	10	70	95	Yes	
stu31	10	10	5	70	10	5	5	70	92.5	Yes	
stu32	0	10	0	35	10	10	10	70	72.5	No	
stu33	10	10	10	35	10	10	10	70	82.5	Yes	
stu34	10	10	10	70	10	10	10	70	100	Yes	
stu35	10	10	10	0	10	10	10	0	30	No	
stu36	5	5	10	35	10	5	5	35	55	No	
stu37	10	10	10	70	10	10	10	70	100	Yes	
stu38	10	5	10	70	10	10	6	70	95.5	Yes	
% of students who submitted both projects and had at least an average grade of 80:									50%		

Appendix 3

Spring 2016 - Two Sections of CIS265											
	Comm-10 Proj2	Prot-10 Proj2	Header-10 Proj2	Stat-70 Proj2	Comm-10 Proj3	Prot-10 Proj3	Header-10 Proj3	Stat-70 Proj3	AVG Score P1&2	80 Or Above	
stu1	0	0	0	0	0	0	0	0	0	No	
stu2	10	10	10	70	10	10	10	70	100	Yes	
stu3	10	10	10	70	10	10	10	70	100	Yes	
stu4	5	5	5	35	10	10	10	70	75	No	
stu5	5	5	10	70	10	10	5	70	92.5	Yes	
stu6	5	10	10	70	10	10	10	70	97.5	Yes	
stu7	0	0	0	0	10	5	10	70	47.5	No	
stu8	10	10	10	35	10	10	10	35	65	No	
stu9	10	10	10	70	10	10	10	0	65	No	
stu10	10	10	10	70	10	10	10	70	100	Yes	
stu11	10	10	10	70	10	10	10	70	100	Yes	
stu12	5	10	5	70	10	5	10	70	92.5	Yes	
stu13	10	10	10	35	5	5	10	70	77.5	No	
stu14	0	0	0	0	0	0	0	0	0	No	
stu15	10	10	10	70	10	10	10	70	100	Yes	
stu16	10	10	10	70	10	5	10	70	97.5	Yes	
stu17	10	10	10	70	10	10	10	70	100	Yes	
stu18	10	10	5	70	10	5	10	70	95	Yes	
stu19	10	10	10	35	10	10	10	70	82.5	Yes	
stu20	10	10	10	70	10	10	10	35	82.5	Yes	
stu21	0	0	0	70	5	0	10	70	77.5	No	
stu22	5	5	5	70	0	0	0	70	77.5	No	
stu23	0	0	0	0	0	0	0	0	0	No	
stu24	10	10	10	70	10	10	10	70	100	Yes	
stu25	10	5	10	70	10	5	10	70	95	Yes	
stu26	5	5	5	70	5	5	10	35	70	No	
stu27	10	10	10	70	10	10	10	70	100	Yes	
stu28	10	10	10	70	5	5	10	35	77.5	No	
stu29	5	10	5	70	0	5	0	70	82.5	Yes	
stu30	0	0	0	70	0	0	0	70	70	No	
stu31	5	5	5	70	10	10	10	35	75	No	
stu32	10	10	0	70	0	10	10	70	90	Yes	
stu33	10	10	10	70	10	10	10	70	100	Yes	
stu34	10	10	10	70	10	10	10	35	82.5	Yes	
stu35	5	5	5	35	5	5	10	35	52.5	No	
stu36	10	10	10	35	10	10	10	35	65	No	
stu37	10	5	5	70	10	10	10	35	77.5	No	
% of students who submitted both projects and had at least an average grade of 80:									54%		