Course Number & Name: CSC 100 Fundamentals of Computer Science

Credit Hours: 3.0  Contact Hours: 3.0  Lecture: 3.0  Lab: N/A  Other: N/A

Prerequisites: Grade of “C” or better in MTH 086 or placement

Co-requisites: MTH 092  Concurrent Courses: None

Course Outline Revision Date: Fall 2010

Course Description: This course introduces the elementary concepts of computer science and is specifically designed for students planning to major in the discipline. The course emphasizes the various aspects of computing such as problem solving, algorithm design, and program construction. Students also explore the application of computer science to various real-world problems. An object-oriented programming language is used to develop the student’s problem solving and programming skills. NOTE: Successful completion of programming projects requires students to use a computer laboratory outside of the class period.

General Education Goals: CSC 100 is affirmed in the following General Education Foundation Category: Technical Competency/Information Literacy. The corresponding General Education Goal is as follows: Students will use computer systems or other appropriate forms of technology to achieve educational and personal goals.

Course Goals: Upon successful completion of this course, students should be able to do the following:

1. demonstrate knowledge of basic hardware logic;
2. explain and interpret internal data representations;
3. implement algorithmic solutions in an object-oriented programming language;
4. use object-oriented design techniques to design algorithmic solutions for a variety of fundamental problems; and
5. discuss the applications of computer science in other disciplines such as business, engineering, medicine, etc.
Measurable Course Performance Objectives (MPOs): Upon successful completion of this course, students should specifically be able to do the following:

1. Demonstrate knowledge of basic hardware logic:
   1.1 identify basic hardware components;
   1.2 describe the functions of hardware components; and
   1.3 explain what problems are non computable

2. Explain and interpret internal data representations:
   2.1 convert decimal numbers into signed, unsigned and 2s complement binary numbers;
   2.2 convert a simple word into ASCII code; and
   2.3 explain the need for Unicode data representation

3. Implement algorithmic solutions in an object-oriented programming language:
   3.1 write an algorithm;
   3.2 translate an algorithm into a high-level programming language; and
   3.3 compile, execute, and test the program

4. Use object-oriented design techniques to design algorithmic solutions for a variety of fundamental problems:
   4.1 use ‘class’ structures to implement a problem solution in an object-oriented program; and
   4.2 execute and test the program

5. Discuss the applications of computer science in other disciplines such as business, engineering, medicine, etc.
   5.1 discuss the use of computer applications in another discipline

Methods of Instruction: Instruction will consist of lectures, laboratory assignments, and programming examples.

Outcomes Assessment: Quiz, test and exam questions are blueprinted to course objectives. Checklist rubrics are used to evaluate the projects for the presence of course objectives. Data is collected and analyzed to determine the level of student performance on these assessment instruments in regards to meeting course objectives. The results of this data analysis are used to guide necessary pedagogical and/or curricular revisions.

Course Requirements: All students are required to:

1. Complete all assigned reading.
2. Participate in class discussions.
3. Complete all assignments on time.
4. Take all quizzes, tests, and exams as scheduled.
Methods of Evaluation: Final course grades will be computed as follows:

<table>
<thead>
<tr>
<th>Grading Components</th>
<th>% of final course grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Attendance and class participation</td>
<td>5%</td>
</tr>
<tr>
<td>Attendance and class participation are necessary for students to benefit from the guidance of the instructor.</td>
<td></td>
</tr>
<tr>
<td>• Homework Assignments (10 – 15)</td>
<td>20%</td>
</tr>
<tr>
<td>Students must practice skills on their own by doing homework to be able to master course objectives. Homework assignments relate to these objectives.</td>
<td></td>
</tr>
<tr>
<td>• Quizzes (dates specified by the instructor)</td>
<td>20%</td>
</tr>
<tr>
<td>Quizzes will provide evidence of student mastery of course objectives.</td>
<td></td>
</tr>
<tr>
<td>• 3 or more Tests (dates specified by the instructor)</td>
<td>25%</td>
</tr>
<tr>
<td>Tests will show evidence of the extent to which students meet course objectives, including but not limited to identifying, explaining, and implementing course content, analyzing and solving problems, and stating appropriate conclusions using correct terminology.</td>
<td></td>
</tr>
<tr>
<td>• Projects</td>
<td>10%</td>
</tr>
<tr>
<td>The same objectives apply as with tests, but it is anticipated that students will provide evidence of synthesizing a combination of concepts.</td>
<td></td>
</tr>
<tr>
<td>• Final Exam (comprehensive)</td>
<td>20%</td>
</tr>
<tr>
<td>The same objectives apply as with tests, but it is anticipated that students will provide increased evidence of synthesizing a combination of concepts.</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: The above ‘% of final course grade’ guide is a suggestion and will vary depending on the instructor. The instructor must inform the students of the specific method of calculation on the class syllabus.

Academic Integrity: Dishonesty disrupts the search for truth that is inherent in the learning process and so devalues the purpose and the mission of the College. Academic dishonesty includes, but is not limited to, the following:

- plagiarism – the failure to acknowledge another writer’s words or ideas or to give proper credit to sources of information;
- cheating – knowingly obtaining or giving unauthorized information on any test/exam or any other academic assignment;
- interference – any interruption of the academic process that prevents others from the proper engagement in learning or teaching; and
- fraud – any act or instance of willful deceit or trickery.
Academic Integrity (continued):

Violations of academic integrity will be dealt with by imposing appropriate sanctions. Sanctions for acts of academic dishonesty could include the resubmission of an assignment, failure of the test/exam, failure in the course, probation, suspension from the College, and even expulsion from the College.

Student Code of Conduct: All students are expected to conduct themselves as responsible and considerate adults who respect the rights of others. Disruptive behavior will not be tolerated. All students are also expected to attend and be on time for all class meetings. No cell phones or similar electronic devices are permitted in class. Please refer to the Essex County College student handbook, Lifeline, for more specific information about the College’s Code of Conduct and attendance requirements.
**Course Content Outline:** based on the text *Computer Science: An Overview* (10th edition) by J Glenn Brookshear: Addison Wesley. **NOTE:** The actual textbook may vary and the amount of time spent on each topic may also vary depending on the class and the instructor.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
</tr>
</thead>
</table>
| 1 – 2 | Introduction  
1. Origin/Study of Algorithms  
2. History of Computing Devices and Contributors to the Field  
3. Abstraction  
4. Social/Ethical/Legal Ramifications |
| 3 – 4 | Data Storage  
1. Main Memory  
2. Mass Storage  
3. Manipulating Bits: Basic Boolean Operations  
4. Representing Functions as Circuits and Truth Tables  
5. Representing Character Data Internally  
6. Representing Numeric Data Internally  
7. Data Communications: Error Checking Methods |
| 5 – 6 | Data Manipulation  
1. Computer Architecture  
2. Machine Language  
3. Program Execution  
4. Arithmetic/Logic Instructions  
5. Communicating with Other Devices  
6. Other Architectures |
| 7 – 8 | Operating Systems  
1. The History of Operating Systems  
2. Operating System Architecture  
3. Coordinating the Machine’s Activities  
4. Handling Competition among Processes  
5. Security | **Test #1**  
**Test #2** |
| 9 – 10 | Networking and the Internet  
1. Network Fundamentals  
2. The World Wide Web  
3. Internet Protocols  
4. Security |
<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
</tr>
</thead>
</table>
| 11 – 13 | **Algorithms**  
   1. The Concept of an Algorithm  
   2. Algorithm Representation  
   **Test #3** (approximately Week 12)  
   3. Algorithm Discovery  
   4. Iterative Structures  
   5. Recursive Structures  
   6. Efficiency and Correctness |
| 14 | **Programming Languages**  
   1. Historical Perspective  
   2. Traditional Programming Concepts  
   3. Procedural Units  
   4. Language Implementation |
| 15 | Review of all course material  
   **Final Exam** on all material covered in the course |

**Note:** Homework and project due dates will be given by the instructor during the semester.