**ESSEX COUNTY COLLEGE**

**Engineering Technologies and Computer Science Division**

**CSC 112 – Computer Programming for Engineering and Technology**

**Course Outline**

**Course Number & Name:** CSC 112 Computer Programming for Engineering and Technology

**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 113 or MTH 119

**Co-requisites:** None **Concurrent Courses:** None

**Course Outline Revision Date:**  Fall 2010

**Course Description**: This course is an introduction to computer-oriented problem solving and programming and their applications in engineering. It provides the essential foundation for a program of study in object-oriented programming and computer-oriented mathematics.  It covers the general areas of data analysis (graphics, sorting, and statistics), curve fitting (regression and interpolation), and equation solving. Students learn programming and the use of general-purpose application software tools such as spreadsheets, database, and mathematical software. Students are required to complete a series of laboratory assignments illustrating applications of computer-oriented problem solving.

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

1. demonstrate knowledge of a high-level programming language;

2. develop algorithms to solve various problems; and

3 write programs (application & applet) using system-defined data types, selection and iteration structures, object-oriented design, and arrays.

**Measurable Course Performance Objectives (MPOs)**: Upon successful completion of this course, students should specifically be able to do the following:

1. Demonstrate knowledge of a high-level programming language:

* 1. *properly use conditional, assignment and loop states*;
  2. *translate verbal expressions into correct language syntax*; and
  3. *compile, test and execute a computer program*

2. Develop algorithms to solve various problems:

2.1 *identify what solutions to a problem constitute an algorithm;* and

2.2 *develop algorithms that will solve specific problems*

**Measurable Course Performance Objectives (MPOs)** (continued):

3. Write programs (application & applet) using system-defined data types, selection and iteration structures, object-oriented design, and arrays:

3.1 *write, document, compile, test and execute a program written in a high-level language that uses data types;*

3.2 *write, document, compile, test and execute a program written in a high-level language that uses selection and iteration structures;*

3.3 *write, document, compile, test and execute a program written in a high-level language that uses object-oriented design;* and

3.4 *write, document, compile, test and execute a program written in a high-level language that uses arrays*

**Methods of Instruction**: Instruction will consist of lectures and computer lab work.

**Outcomes Assessment:** Test and exam questions are blueprinted to course objectives. Rubrics are used to evaluate the programming assignments and tests for evidence of mastery of the content of the course. 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. Be on time for all scheduled classes.

2. Read the assigned textbook.

3. Do text book problems/homework as assigned.

4. Perform all laboratory assignments and maintain a record of data.

**Methods of Evaluation:** Final course grades will be computed as follows:

**% of**

**Grading Components final course grade**

* **Attendance 10%**

Attendance is necessary for students to benefit from the guidance of the instructor.

* **2 or more Tests** (dates specified by the instructor)  **30%**

Tests will show evidence of the extent to which students are able to write problem solutions in the proper syntax of a high-level language, which is related to course objectives.

**Methods of Evaluation** (continued)**:**

**% of**

**Grading Components final course grade**

* Homework (Projects/Programming Assignments) 30%

Students must practice skills on their own by doing homework to be able to master course objectives. Homework and programming assignments relate to these objectives.

* **Final Exam** **30%**

The **comprehensive** final exam will examine the extent to which students can write algorithms and implement those solutions in a high level language. It is anticipated that students will provide increased evidence of synthesizing a combination of concepts covered in the course.

**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.

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 **Java Software Solutions**, 6th edition, by Lewis & Loftus; published by Prentice Hall, 2008; and using the required software jGRASP (java development toolkit).

**Week Topic Reading Assignments**

|  |  |  |
| --- | --- | --- |
| 1 | Java Programming Language  Program Development  Object-Oriented Programming | Chapter 1 |
| 2 | Character Strings  Variables & Assignment | Chapter 2 |
| 3 | Primitive Data Types/Expressions  Data Conversion/ Interactive Programs | Chapter 2 (continued) |
| 4 | Creating Objects/ Packages  The String, Random, and Math Classes | Chapter 3 |
| 5 | Formatting Output/Enumerated Types  Wrapper Classes  **Test 1** | Chapter 3 (continued) |
| 6 | Anatomy of a Class  Encapsulation | Chapter 4 |
| 7 | Anatomy of a Method  Constructors | Chapter 4 (continued) |
| 8 | Boolean Expressions  If-statement & switch-statement | Chapter 5 |
| 9 | While, do, and for statements  Iterators | Chapter 5 (continued) |
| 10 | Identifying Classes & Objects  Class Relationships | Chapter 6 |
| 11 | Interfaces/ Method Design/ Method Overloading | Chapter 6 (continued) |
| 12 | **Test 2**  Array Elements/Array of Objects  Command Line Arguments | Chapter 7 |
| 13 | Variable Length Parameter Lists  Two-Dimensional Arrays | Chapter 7 (continued) |
| **Week** | **Topic** | **Reading Assignments** |
| 14 | The ArrayList Class | Chapter 7 (continued) |
| 15 | Course Review  **Final Exam** |  |

Note: Homework (projects/programming assignments) is assigned on an ongoing basis throughout the semester to correspond to the topics being discussed in class.