

**Department Master Syllabus
Camden County College
Blackwood, New Jersey**

Course Title: **Fundamentals of Programming**

Course Number: **CSC-105**

Department/Program Affiliation: **Computer Science**

Date of Review:

(This Department Master Syllabus has been examined by the program/department faculty members and it is decided that no revision is necessary at this time.)

Date of Last Revision: **February 2017**

(This Department Master Syllabus has been examined by the program/department faculty members and it is decided a change requiring a revision is necessary at this time.)

N.B. A change to the course materials alone (textbooks and/or supplementary materials) may not constitute a revision. Any other change to the items listed below on this form is considered a revision and requires approval by the program faculty at a Program/Department Meeting and by the division at a Chairs and Coordinator Meeting.

Credits: **4**

Contact Hours: **Lecture 4 Lab 0 Other 0**

Prerequisites: **None**

Co-requisites: **None**

Course Description: This is an introductory Computer Science course in which students will learn the fundamentals of problem solving and object-oriented programming in a 3-dimensional, interactive, animation environment. Students will create animation projects using a special software package for creating animation in small virtual worlds using 3-dimensional models. Students will obtain a strong core of fundamental programming concepts and problem-solving techniques, providing a basis for further study in a variety of computer related fields.

Course Student Learning Outcomes:

Upon completion of this course, the student will be able to:

- Discuss the fundamental stages of software development, the life cycle of a software system, and the importance of system testing as assessed by graded exercises, homework assignments, programming projects, presentations, and exams.
- Describe the terms, concepts and problem-solving techniques of an object-oriented approach to software design/development including Classes, Objects, Inheritance, and Composition as assessed by graded exercises, homework assignments, presentations, and exams.
- Develop and implement programming solutions through the process of top-down, stepwise refinement using selection, repetition, sequential control structures, and recursive techniques where appropriate as assessed by graded exercises, homework assignments, programming projects, presentations, and exams.

- Construct programs with methods and pass parameters to those methods as assessed by graded exercises, homework assignments, programming projects, presentations, and exams.
- Define and use fundamental data types, lists, and arrays for basic data organization as assessed by graded exercises, homework assignments, programming projects, presentations, and exams.
- Create Event Listeners and write the appropriate code to handle events as assessed by graded exercises, homework assignments, programming projects, presentations, and exams.
- Modify a project originally created in a 3-d animation environment, using a professional object-oriented development environment as assessed by graded exercises, homework assignments, programming projects, presentations, and exams.

General Education Student Learning Outcomes (if applicable):

Upon completion of this course, the student will be able to:

- Use computer systems and/or other appropriate forms of technology to present information as assessed by graded exercises, homework assignments, programming projects, presentations, and exams.
- Use appropriate forms of technology to identify, collect, and process information as assessed by graded exercises, homework assignments, programming projects, presentations, and exams.

Course Outline:

1. Introduction to Programming and Problem Solving
2. Introduction to a 3-dimensional Programming Environment
3. Objects, Classes and API's
4. Design and Algorithm Development
5. Implementation and Testing
6. Methods, Variables and Expressions
7. Parameters
8. Inheritance
9. Program Control
 - Use random numbers to implement simple simulations
 - Data types
 - Conditionals
 - Repetition and Recursion
10. Event-driven Programming Concepts
 - Interactive Programs
 - Event-listener procedures/methods
 - Event-handler procedures/methods
11. The Array Data Structure
 - Creating, using, and iterating through arrays of objects
 - Using Iterators with arrays
 - Searching for an element within an array
12. Programming and Problem Solving using a professional integrated development environment
 - Write, evaluate and debug a code fragment that calls the relevant methods of a class to solve a problem based on the class's API.
 - Evaluate and modify a class using a professional programming language

Course Activities: The classroom activities will include formal and informal lectures where new material and assigned problems will be explained. Students will have the opportunity to contribute to the discussion and to ask questions about the material. Hands-on computer activities will take place to reinforce concepts presented during the lecture. Projects are assigned to be completed outside of the regularly scheduled class time.

Assessment of Student Learning Outcomes: The student will be evaluated on the degree to which student learning outcomes are achieved. A variety of methods may be used such as tests, class participation, projects, homework assignments, etc.

Course Materials:

Textbook(s): To be determined

Supplemental Materials: Announced first day of class as needed.