

# Department Master Syllabus

**Camden County College**

**Blackwood, New Jersey**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Number:**  CSC-122 | | **Course Title:**  Computer Science I | | | |
| **Department/Program:** Computer Science | | | | | |
| **Date of Review:** Click here to select a month. | | Click here to select a year. | | | |
| (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 Revision:** December | | | | 2021 | |
| (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 department/program faculty at a department/program meeting and by the division at a Chairs and Coordinator meeting. | | | | | |
| **Credits:** 4 | | | | | |
| **Contact Hours** | **Lecture:** 3 | | **Lab:** 3 | | **Other:** 0 |
| Prerequisites: CSC-121 | | | | | |
| Co-requisites: None | | | | | |
| Course Description: This course emphasizes problem-solving strategies, analysis of algorithms and the use of simple data structures to formulate object-oriented programming solutions to problems. Topics covered include: primitive data types, constructors, visibility, control structures, classes, methods, inheritance, polymorphism, arrays, Strings, and packages that make up the Java API. Object-oriented concepts and terminology will be presented with a focus on using classes for program specification and design. The concept of an Abstract Data Type is presented. The student will learn how classes are declared, defined, used and organized into coherent designs. Students will apply the concepts through programming assignments in an object-oriented language. Data abstractions, information hiding, software reusability and extendibility will be stressed. During labs students will explore data abstractions, information hiding, software reusability and extendibility. | | | | | |
| **Student Learning Outcomes (SLOs)**  Course specific student learning outcomes  Upon completion of this course the student will be able to:   1. Examine the issues of large-scale software development projects including the ramifications of security, the importance of professionalism and ethical considerations of computing professionals as assessed by graded exercises, homework assignments, programming projects, and exams. 2. Describe the terms, concepts and techniques of an object-oriented approach to software design/development and compare and contrast this approach to top-down, structured design methodology as assessed by graded exercises, homework assignments, programming projects, and exams. 3. Identify objects associated with a stated problem, then design, code, and test a set of classes for their representation that demonstrates encapsulation, data abstraction, and information hiding using simple ADTs as assessed by graded exercises, homework assignments, programming projects, and exams. 4. Formulate programming solutions that use inheritance, polymorphism and class hierarchies to promote reuse of program design and code as assessed by graded exercises, homework assignments, programming projects, and exams. 5. Design, implement, test and debug programs that maximize the use of existing library functions and classes as assessed by graded exercises, homework assignments, programming projects, and exams. 6. Organize programs based on separate declaration, definition, and usage files that demonstrate security considerations as assessed by graded exercises, homework assignments, programming projects, and exams. 7. Develop applications based on code designed for reuse and extensibility using best practice software engineering techniques as assessed by graded exercises, homework assignments, programming projects, and exams.   As assessed by:  The student will be evaluated on the degree to which student learning outcomes are achieved. A variety of methods may be used such as exams, class participation, projects, homework assignments, online learning tools, etc. | | | | | |
| **General Education Student Learning Outcomes**  If this course has applied for General Education Elective Status the general education student learning outcomes listed below must exactly match those the sponsor has identified on the General Education Request form.  General Education SLOs:  N/A  As assessed by:  N/A | | | | | |
| **Program Learning Outcomes**  List all course level student learning outcomes that interconnect to a particular program learning outcome.  At the end of the program, the graduate will be able to:   * 1. Analyze, design, develop and test computer-based applications using problem-solving and analytical skills developed throughout the program. PSLO-CSC.AA #1; SLO # 1-7   CSC.AS #1; SLO # 1-7   * 1. Apply scientific and mathematical principles to study computer science. PSLO-CSC.AS #3; SLO # 1-7   Describe the assessment of the interconnected program learning outcome(s).  The student will be evaluated on the degree to which student learning outcomes are achieved. A  variety of methods will be used such as tests, class participation, projects, homework assignments,  etc. | | | | | |
| **Course Outline:**  1. Introduction  2. Data and Expressions  3. Using Classes and Objects  a. String Class  b. Random Class  c. Math Class  d. Enumerated Types  e. Wrapper Classes  4. Conditionals and Loops  5. Writing Classes  a. Encapsulation  b. Class Relationships  c. Constructors  d. Method Design  e. Testing  6. Graphical User Interfaces  a. GUI Components  b. Layout Managers  c. Events and Event Handlers  7. Arrays  a. Declaring and using Arrays  b. Arrays of Objects  c. Multidimensional Arrays  8. Inheritance  a. Creating Subclasses  b. Overriding Methods  c. Class Hierarchies  d. Visibility  9. Polymorphism  a. Dynamic Binding  b. Interfaces  10. Exceptions  a. Exception Handling  b. Exception Propagation  c. Checked vs. Unchecked Exceptions | | | | | |
| **Course Activities:**  The classroom activities will include formal and informal lectures and lab sessions. During lectures, new material and assigned problems will be explained. Students are encouraged to contribute to the discussion and to ask questions about the material. Structured, supervised, "hands-on" lab sessions are built-in to the semester schedule. These sessions will be structured to amplify lecture material. Attendance during lab sessions and completion of the laboratory assignments are mandatory. All programming project assignments are to be completed outside of the regularly scheduled labs and class meetings. | | | | | |
| **Course Materials:**  Textbook(s): TBD  Supplemental Materials: Click here to enter text or N/A  Software Licenses: All software required is open source.  Computers: Students will need access to a computer for assigned projects and homework. | | | | | |
| **Course Assessment Plan**  How often and by what means will the effectiveness of this course as part of the curriculum be assessed?    Assessment cycle to be determined by the members of the department. Students will be evaluated on the degree to which student learning outcomes are achieved. Assessment instruments may be in the form of tests and/or programming projects. | | | | | |