

# Department Master Syllabus

**Camden County College**

**Blackwood, New Jersey**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course Number:**  CSC-171 | | **Course Title:**  Introductory Python Programming | | | |
| **Department/Program:** 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 Revision:** November | | | | 2023 | |
| (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:** 3 | | | | | |
| **Contact Hours** | **Lecture:** 3 | | **Lab:** 0 | | **Other:** 0 |
| Prerequisites: None | | | | | |
| Co-requisites: None | | | | | |
| Course Description: In this course, students are introduced to programming and problem solving using the  Python language. Algorithm development and basic problem solving techniques are introduced using a procedural approach. Topics covered include programming with numbers, strings, lists, tuples, sets, dictionaries, files, control structures, functions with parameter passing, scope and the Python libraries. Finding and fixing errors, using a debugger and an introduction to exception handling, are presented and implemented. | | | | | |
| **Student Learning Outcomes (SLOs)**  Course specific student learning outcomes  Upon completion of this course the student will be able to:   1. Select and implement appropriate data types, expressions and control structures including sequence, selection, and repetition to solve simple programming problems. This course learning outcome will be assessed using a variety of methods including problem sets, assigned programs, and/or exam questions. 2. Design algorithms, create, document, and test programming solutions that read and process numeric and string inputs, data collections, and files that produce correct output and results. This course learning outcome will be assessed using assigned programming problems. 3. Develop techniques for decomposing complex tasks into reusable functions that exchange and share information via parameter passing and returned values. This course learning outcome will be assessed using a variety of methods including problem sets, assigned programs, and/or exam questions. 4. Determine when to use strings, lists, dictionaries, tuples, and sets and describe their operations. This learning outcome will be assessed using a variety of methods including problem sets, assigned programming exercises, and/or exam questions. 5. Describe and analyze programming code and errors, trace execution and perform validity testing and debugging. This learning outcome will be assessed using a variety of methods including problem sets, assigned programming exercises, and/or exam questions. 6. Analyze and manage exceptions and exception handling in programs. This learning outcome will be assessed using a variety of methods including problem sets, assigned programming exercises, and/or exam questions.   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, programming 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. Develop solid analytical reasoning, critical thinking and technical skills in order to extract, wrangle, analyze and present data for multiple disciplines to broad audiences that follow professional standards to enhance understanding and decision- making. PSLO-DSC.AAS #1; SLO # 1-6   2. Demonstrate the ability to work independently and as a member of a team with modern technical tools to accomplish data life cycle project goals and meet deadlines. PSLO-DSC.AAS #1; SLO # 1-6   Describe the assessment of the interconnected program learning outcome(s).  Various course level assessment instruments will be used to target specific program learning outcomes. | | | | | |
| **Course Outline:**   1. Introduction to High-Level Programming and the Python Language    1. Gaining familiarity with the programming environment    2. Python Interpreter    3. Algorithm design    4. Converting an algorithm into a Python program 2. Programming Fundamentals: Variable Types and Values    1. Numeric and string data types, variables and constants    2. Performing arithmetic    3. Input and output    4. Documenting programs 3. Decision Control Structures    1. Making decisions using If statements    2. Relational and logical operators    3. Boolean variables    4. Nested decisions and multiple alternatives    5. Performing input validation and testing branches and conditions 4. Repetition Control Structures    1. Common loop algorithms    2. Conditional loops: While loop and sentinels    3. Counting loops: For loop, iterating, and range function    4. Nested loops    5. Tracing, testing and debugging programs that use control structures 5. Introduction to Python Standard Libraries 6. Functions    1. Implementing functions as reusable components    2. Writing and documenting function definitions    3. Passing parameters    4. Returning values    5. Problem solving: Stepwise Refinement    6. Variable Scope    7. Introducing Randomness 7. Built-in Python Data Structures    1. Comparing data structures    2. Syntax    3. Use cases    4. Converting between data structures 8. List and Tuples    1. Common uses and operations of Lists    2. Common List algorithms    3. Common uses and operations of Tuples    4. Comparing Lists with Tuples    5. Using Tables as a List of Lists 9. Sets and Dictionaries    1. Common uses and operations of Sets    2. Comparing Sets with Lists    3. Common uses and operations of Dictionaries    4. Comparing Dictionaries with Lists 10. Files and Exception Handling     1. Reading and Writing text files     2. Exception Handling and managing runtime errors | | | | | |
| **Course Activities:**  The classroom activities will include formal and informal lectures where students can ask questions and participate in directed active-learning lessons. During lectures, new material and assigned problems are explained. Students are encouraged to contribute to the discussions and to ask questions about the material presented. Active-learning lessons include guided problem solving and best practice programming techniques using applicable programming problems. | | | | | |
| **Course Materials:**  Textbook(s): TBD  Supplemental Materials: TBD  Software Licenses: Free software tools  Computers: Will be needed by online students and used for instruction, student learning activities and homework in face-to-face courses. | | | | | |
| **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. | | | | | |