CSC-125 Python Programming

Credits: 3 | **Semester:** Fall 2025

In-Person Sessions: Mondays, 10:00 AM - 11:15 AM | Charlestown Campus, Building D, Room D102A

Asynchronous Components: Online coursework designed for mid-week completion will be distributed through BHCC Moodle

Instructor: Johnathan Oneal | **Email:** johnathan.oneal@bhcc.edu

Course Description

This is an introductory course designed for any student interested in learning computer programming concepts and hands-on computational thinking, all in the context of the Python programming language. No prior programming experience is necessary. Students will use their problem-solving abilities to implement programs in Python. This course will show students how to create basic programming structures including decisions and loops. Furthermore, students will explore unique Python data structures such as tuples and dictionaries. Students will also learn to perform basic debugging techniques. At the end of this course, students will have learned enough computer science and programming concepts to write Python programs to solve problems on their own. This course will prepare students to move on to the Advanced Python Programming course.

Learning Objectives

By the end of this course, you will be able to:

- Create and execute Python programs using appropriate syntax and structure
- Declare and manipulate variables of different data types (strings, integers, floats, booleans)
- Implement decision-making logic using conditional statements (if, elif, else)
- Design and implement repetitive processes using for and while loops
- Create and call functions with parameters and return values
- Work with Python data structures including lists, tuples, and dictionaries
- Import and utilize Python standard library modules
- Implement basic error handling using try/except blocks
- Apply systematic debugging techniques to identify and resolve programming errors
- Design and complete an independent programming project from concept to implementation

Required Materials

Platform: Kaggle Notebooks (free cloud-based platform - no software installation required)

Textbooks and Resources: This course is designed to be completely free and accessible to all students. With the wealth of high-quality programming resources available online, we will use carefully selected free tutorials, documentation, and learning materials throughout the semester. All required readings and supplemental materials will be posted to the course site one week in advance, ensuring you have adequate time to access and review the content. No textbook purchases are required—the goal is to eliminate any possible barriers while providing excellent educational resources.

Students are **strongly** encouraged to seek out additional learning materials beyond AI-generated content. While AI tools can be helpful for quick answers, they are not designed as structured teaching resources. Developing the ability to identify and engage with high-quality instructional materials, particularly those aligned with your specific interests and learning goals, is a crucial skill for lifelong learning in programming and technology.

Primary Resource Sources:

- LearnPython.org - Interactive programming tutorials and exercises
- Python 101 (python101.pythonlibrary.org) - High-quality Python reference and examples
- Full Stack Python (fullstackpython.com) - Professional applications and career context
- Official Python documentation and community resources

Hardware Requirements:

- Computer or tablet with reliable internet access
- Web browser (Chrome, Firefox, Safari, or Edge)

Teaching Methods

This course uses a **hybrid approach** combining in-person instruction with asynchronous online work. **Monday classes are in-person** for live instruction, demonstrations, and periodic quizzes. The remainder of each week is **asynchronous distance learning** with materials uploaded a week in advance through Moodle, allowing you to work at your own pace while meeting weekly deadlines.

Weekly Schedule:

- **Monday:** In-person class for new concept instruction and periodic assessments
- **Wednesday:** Expected completion date for asynchronous work (self-paced target)
- **Friday:** Homework assignment deadline

You'll import pre-designed .ipynb notebook files containing exercises and complete assignments directly in the cloud environment. The course emphasizes learning by doing, with extensive practice exercises and real-world applications.

Office Hours are available by appointment through email.

Student Success Philosophy

I believe in rewarding effort and genuine engagement with the material. I am committed to helping students who demonstrate they are putting in the work and making an honest attempt to learn. However, I cannot provide meaningful assistance to students who consistently submit late work or assignments that show little to no effort. Students are encouraged to reach out if they are struggling or have questions about the course material. I also value and encourage curiosity—students who ask thoughtful questions, explore beyond the basic requirements, or show genuine interest in learning will find that enthusiasm recognized and supported.

Grading and Assessment

Grade Distribution

Component	Weight	Description
Homework Assignments	30%	Weekly notebook exercises (two lowest scores dropped)
Pseudocode/Program Design	10%	Two algorithm design assignments
Quizzes	20%	Periodic in-person assessments (lowest two scores dropped)
Midterm Exam	20%	In-person paper exam (Week 10)
Final Project	20%	Independent programming project

Grading Scale

Grade	Numerical Range
A	94 - 100
A-	90 - 93
B+	87 - 89
B	83 - 86
B-	80 - 82
C+	77 - 79
C	70 - 76
D	60 - 69
F	0 - 59

Assignment Details

Homework Assignments (30%):

- Due date will be in Moodle but usually Fridays at 11:59 PM (except weeks with pseudocode projects)
- Submitted via file upload through Moodle
- **Two lowest homework scores will be dropped**
- **Late Policy:** 10% penalty per day for up to three days, no submissions accepted after three days

Pseudocode/Program Design Projects (10%):

- Two algorithm design assignments replacing traditional homework
- Week 6: Design algorithm for list manipulation problem using pseudocode

- Week 12: Design algorithm for the final project (data processing using dictionaries) in pseudocode
- Focus on problem-solving and logical thinking before coding
- Due date will be in Moodle but usually Friday by 11:59 PM, will replace the homework assignment for the week
- **Late Policy:** Same as homework (10% per day for up to three days, then zero)

Quizzes (20%):

- Administered during Monday in-person classes periodically throughout the semester
- Cover material from previous weeks
- Focus on conceptual understanding and code reading
- Designed to ensure students are engaging with distance education materials appropriately and completing the homework. These are designed not to be challenging if you complete all the materials and homework for the previous week
- **Lowest two quiz scores will be dropped**
- No make-up quizzes except for excused absences

Midterm Exam (20%):

- Covers material from Weeks 1-9
- Monday, November 3rd, 2025 (tentative) during regular class time
- Mix of multiple choice, short answer, and code writing (by hand)

Final Project (20%):

- Overarching programming project combining all course concepts
- Includes planning documentation and working code
- Technical design due Week 12
- Final project due Week 15

Detailed Course Schedule

Week	Topic	Programming Concepts	Assessment
1	Setup, Intro, Syllabus Review, Python Intro	Course overview, syllabus, Python introduction, basic concepts	Homework
2	Variables and Data Types	Variables, int/float/str/bool, type conversion, input(), arithmetic	Homework
3	String Operations & Debugging	String methods, slicing, f-strings, error messages, debugging	Homework
4	Math Operations and Operators	Arithmetic/comparison/assignment operators, boolean expressions, precedence	Homework
5	Conditional Logic	if/elif/else, logical operators, nested conditions, boolean algebra	Homework
6	Lists and Tuples	Creating sequences, indexing, slicing, mutable vs immutable, basic methods	PSEUDOCODE
7	Advanced List Operations	List methods, copying, split()/join(), list comprehensions	Homework
8	For Loops	For loop syntax, range(), nested loops, enumerate(), iteration patterns	Homework
9	While Loops & Advanced Debugging	While loops, break/continue, input validation, debugging strategies	Homework
10	Midterm + Project Planning	Exam (Weeks 1-9), software development lifecycle, project scoping	**MIDTERM**
11	Functions	def keyword, parameters/arguments, return values, scope, docstrings	Homework
12	Advanced Functions & Dictionaries	Default parameters, lambda functions, dictionaries, key-value pairs	**PSEUDOCODE**
13	Libraries and Modules	Import statements, random/math/datetime modules, documentation	Homework
14	Error Handling	try/except blocks, exception types, debugging methodology	**FINAL PROJECT**
15	Final Project	Project integration, documentation, testing, code quality	**FINAL PROJECT**

Course Policies

Academic Integrity

Students are expected to uphold the highest standards of academic honesty throughout this course. All work submitted must represent your own understanding and effort. While collaboration and discussion with classmates are encouraged for learning purposes, all submitted assignments must be your own individual work. Copying code from classmates, online sources, or any other external source without proper attribution constitutes academic dishonesty.

Students who violate academic integrity policies may receive a failing grade for the assignment or course, depending on the severity of the violation.

AI Usage Policy

Regardless of what tools or resources you use to assist in your learning, you are responsible for writing and understanding all code you submit. You must be able to explain any code you turn in and demonstrate your comprehension of how it works. If you cannot explain how your submitted code functions, it will be considered a violation of academic integrity.

Attendance

Regular attendance at Monday in-person classes is required for live instruction and periodic assessments. The Bunker Hill Community College Attendance Policy is as follows: Because poor attendance generally results in poor grades, students must attend all regularly scheduled classes and laboratory sessions. Students having attendance difficulties should discuss this matter with their course instructors and advisors.

Students who know they will be absent for three or more consecutive class sessions because of a family emergency or personal illness should report the extended absence to the Office of the Dean of Students. The Associate Vice President will notify the course instructors of the absence. Upon returning to class, students are responsible for discussing completion of all course requirements with their course instructors. If the course instructor and student find it impossible to complete all assigned work, the student may need to withdraw from the course.

Communication

Primary communication via email and the course Moodle. Check email and course announcements regularly.

Accommodations

Academic accommodations are handled through the Disability Support Services office. Students who have questions about how to register with the Disability Support Services Office can walk in to E222 to meet with an available DSS staff member or can call or email to schedule an appointment. To schedule an appointment, contact Disability Support Services at disabilitysupport@bhcc.edu or 617-228-2327.

Distance Education Information

Course Delivery

Hybrid format - Monday in-person classes with asynchronous online work throughout the week

Materials Distribution

All asynchronous learning materials will be distributed through BHCC Moodle one week in advance of the target completion date. This includes:

- Weekly instructional content and tutorials
- Pre-designed .ipynb notebook files for exercises
- Supplemental reading materials and resources
- Assignment instructions and rubrics
- Video demonstrations and coding examples
- Practice exercises and self-assessment tools

Technology Requirements

- Stable internet connection for asynchronous work and Moodle access
- Web browser capable of running Kaggle Notebooks and accessing Moodle
- Email access for course communications

This syllabus is subject to change. Students will be notified of any modifications via course announcements.