CSC141-A: Foundations of Computer Science I/Lab

Fall 2024

Professor:
Dave Kaul
Cell: 912.844.4595
dkaul@albright.edu

Your Teaching Assistants:
8AM labs have Clinton Odor adimchimma.odor001@albright.edu
9:30AM labs have Wilfredo Vazquez wilfredo.vazquez001@albright.edu

Course Overview

This course is an introduction to Python programming, focusing on foundational concepts and practical application through project-based learning. Over the course of 15 weeks, students will develop their skills in Python by working through the "Python Crash Course, 3rd Edition" textbook. The course will culminate in a final project based on the Aliens game, as described in chapters 12-14.

Important:

If you have prior programming experience and find the assignments aren't challenging enough, please complete the required assignments, including Alien Invasion. Afterward, you can take on an additional Python project to create an AI-powered chatbot as a final task to further enhance your learning.

I don’t want any student to feel bored, and I’ve had a student tell me that the class was too easy. I want to ensure that everyone is both challenged and engaged, so please take advantage of this opportunity to push your skills further.

Course Materials

- **Textbook:** *Python Crash Course, 3rd Edition* by Eric Matthes. I will also provide you with a pdf copy of the textbook, with the agreement that you will buy the book itself.

- **Platform:** GitHub will be used for collaboration and storing class materials. Canvas will be used for grades, attendance, and announcements.

Grading Breakdown

- **Try It Yourself Assignments (Weeks 1-7, 9-11):** 50%
- **Code Jam (Week 8):** 20%
- **Final Alien Invasion Project (Weeks 12-14):** 30%

Course Schedule

Week 1: Introduction & Getting Started

- **Chapters:** 1 - Getting Started
- **Topics:** Python environment setup, basic syntax, running Python programs
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 1

Week 2: Variables and Data Types

- **Chapters:** 2 - Variables and Simple Data Types
- **Topics:** Variables, strings, numbers, comments
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 2

Week 3: Introducing Lists

- **Chapters:** 3 - Introducing Lists
- **Topics:** Creating, modifying, and looping through lists
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 3

Week 4: Working with Lists

- **Chapters:** 4 - Working with Lists
- **Topics:** List comprehensions, slicing, sorting, and copying lists
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 4

Week 5: If Statements

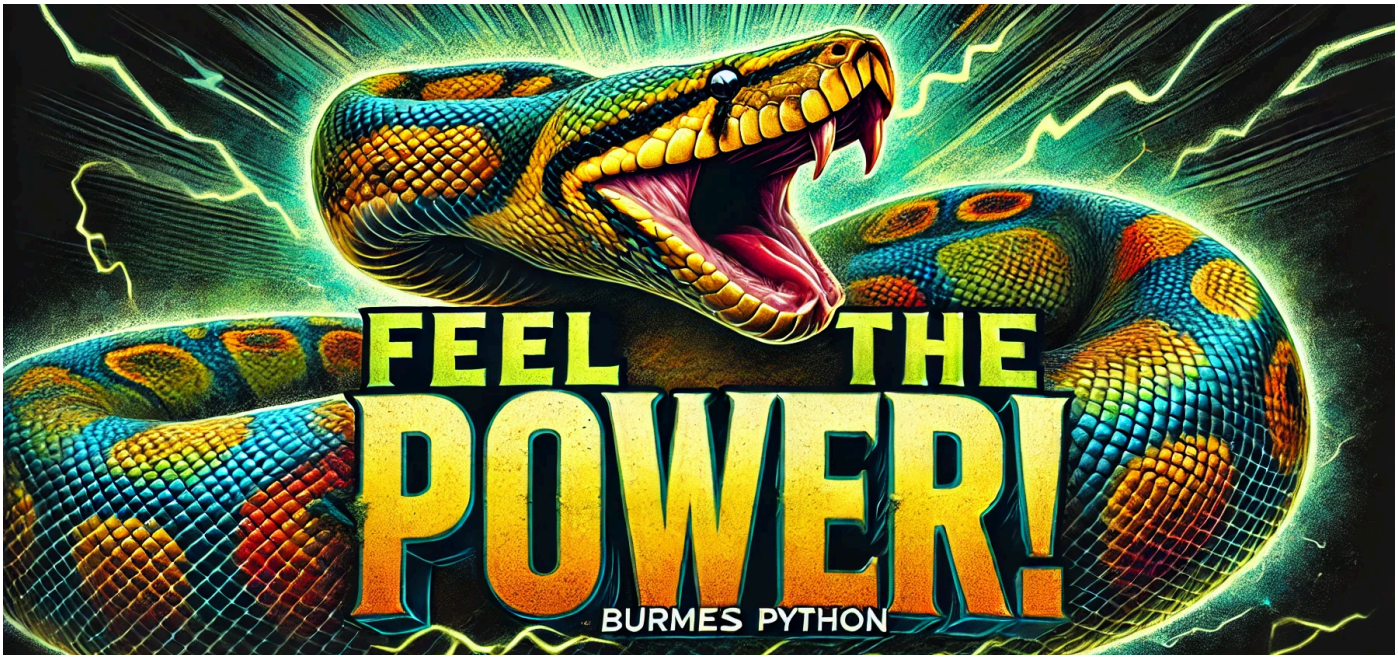
- **Chapters:** 5 - If Statements
- **Topics:** Conditional tests, if-else statements, nesting conditions
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 5

Week 6: Dictionaries

- **Chapters:** 6 - Dictionaries
- **Topics:** Creating and modifying dictionaries, looping through key-value pairs
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 6

Week 7: User Input and While Loops

- **Chapters:** 7 - User Input and While Loops
- **Topics:** Handling user input, using while loops for repetitive tasks
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 7



Week 8: Code Jam - October 16th to the 20th (After Fall Break, although you can work throughout the break if you'd like)

Read and Complete Chapters: 15, 16, 17 - Generating and Downloading Data, Working with APIs

Chapter 15: Generating Data

Learn how to generate data sets and visualize them using Python.
Explore different types of plots and graphs to represent data meaningfully.

Chapter 16: Downloading Data

Understand how to access and download data from the web using Python.
Work with real-world data sets and learn how to parse and clean data for analysis.

Chapter 17: Working with APIs

Learn how to interact with APIs to fetch data dynamically.
Gain experience in authenticating and retrieving data from various web services.

Activity: Code Jam

Format: This is an intensive coding session where you'll apply the concepts learned in Chapters 15, 16, and 17.

Teamwork: You can choose to work solo or partner with one other student. Collaboration is encouraged to enhance learning and tackle more complex projects.

Project: The objective is to create a project that involves generating, downloading, and visualizing data, incorporating API usage if applicable. This project should demonstrate your ability to apply the skills from these chapters in a practical, creative way.

Guidelines:

Scope: Your project should be substantial enough to reflect a full understanding of the chapters but manageable within the week.

Innovation: Try to come up with a unique application or visualization that shows creativity and deeper engagement with the material.

Documentation: Ensure your code is well-documented, including comments that explain your logic and approach. Consider including a `README.md` file in your GitHub repository that outlines the project goals, the data sources, and how to run the code.

Submission: GitHub Repository

Due Date: The project is to be completed and submitted by the end of Week 8.

Repository: Create a new GitHub repository specifically for this project.

Naming: Name the repository following the format:
`codejam_week8_[yourname]` or `codejam_week8_[teamnames]`.

Evaluation:

Your project will be evaluated based on creativity, functionality, proper use of APIs, data handling, code quality, and documentation.

You are to make innovative use of what you have learned in these chapters and create something more powerful.

This project will account for 20% of your overall grade, so take this opportunity to showcase your understanding and skills.

Week 9: Functions

- **Chapters:** 8 - Functions
- **Topics:** Defining functions, passing arguments, return values, working with modules
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 8

Week 10: Classes

- **Chapters:** 9 - Classes
- **Topics:** Object-oriented programming, creating classes, inheritance
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 9

Week 11: Files and Exceptions

- **Chapters:** 10 - Files and Exceptions
- **Topics:** Reading from and writing to files, handling exceptions
- **Assignment:** Complete all "Try It Yourself" exercises from Chapter 10

Week 12: Final Project - A Ship That Fires Bullets

- **Chapters:** 12 - A Ship That Fires Bullets
- **Topics:** Introduction to Pygame, creating a simple game
- **Project Part 1:** Implement the game as described in Chapter 12

Week 13: Final Project - Aliens!

- **Chapters:** 13 - Aliens!
- **Topics:** Extending the game with more features, adding complexity to gameplay
- **Project Part 2:** Continue building the game as described in Chapter 13

Week 14: Final Project - Scoring

- **Chapters:** 14 - Scoring
- **Topics:** Adding a scoring system, tracking player stats
- **Project Part 3:** Finalize the game with scoring features as described in Chapter 14

Week 15: Final Project Presentations & Submission

- **Activity:** Students will present their final projects, demonstrate their code, and discuss their development process.

- **Final Submission:** Complete GitHub repository with all parts of the final project.

Week 16: Reflection on your learning

- There will be no final exam for this course. Students are encouraged to use this time to reflect on their learning and complete any outstanding work. Class WILL meet during the 2 hour time block and students must attend this final class.

Additional Information

- **Office Hours:** Available for one-on-one help with assignments or projects. Please schedule in advance.
- **Collaboration:** Encouraged through GitHub. Review your peers' work and provide constructive feedback.
- **Attendance:** Track through Canvas; please check regularly for any announcements.

Naming Conventions

1: Go to github.com

You will submit every assignment in your GitHub repository

2: GitHub Account Naming Convention:

- Format**:** FirstName-LastName
- Example: If a student's name is Jane Doe, their GitHub account name should be Jane-Doe.
- Use a recent selfie of you for your profile picture
- Write a bio (ChatGPT is good for this)

Additional Guidelines:

- Capitalization**:** Capitalize the first letter of their first and last names for readability (e.g., John-Smith).
- Hyphenation**:** If a student has a hyphenated last name, they should keep the hyphen (e.g., Mary-Smith-Jones).
- No Spaces or Special Characters: Only use hyphens to separate names, and avoid spaces or special characters.

3: Create a Repository:

- Repository Name: After creating your GitHub account, create a new repository with the following name: `pcc_3e`

4: Set Up the Repository Structure:

- Folder Structure: Inside your repository, create a folder for each assignment using the following format:
  - `01_getting_started`

    - `02_variables_and_data_types`
    - `03_introducing_lists`
    - (Continue for each chapter)
-

5: Make the Repository Public

- Ensure that your repository is public so that it can be accessed by Professor Kaul for grading purposes.

5. Share the Repository Link:

- After setting up your repository, share the link at: CSC_141_Fall_2024_Student_Info

5: Assignment Naming Convention:

- Upload Assignments: Inside each folder, upload your code files and a `README.md` file that briefly explains the assignment.

1. Assignment 1-1: 01_01_python_org.py
    - Exercise 1-1: python.org
    - Description: Visit the Python website and explore its resources. Write a brief summary of what you found.
2. Assignment 1-2**:** 01_02_hello_world_typos.py
    - Exercise 1-2: Hello World Typos
    - Description: Create a simple Python script with intentional typos and troubleshoot the errors.
3. Assignment 1-3: 01_03_infinite_skills.py
    - Exercise 1-3: Infinite Skills
    - Description: Reflect on your previous programming experiences and how they might help you in this course

    (Continue for each chapter)


CSC 141
CSC141
BURMES PYTHON
GET EXTREME!
-GET EXTREME!-
CSC 141

General Assignment Grading Rubric

Criteria	Full Credit (100%)	Partial Credit (70-90%)	Minimal Credit (50-69%)	No Credit (0-49%)
Completion	All tasks and requirements are fully completed.	Most tasks and requirements are completed, with minor omissions.	Some tasks or requirements are incomplete or missing.	Very few or no tasks or requirements are completed.
Functionality	Code runs correctly with no errors and produces the expected results.	Code runs with minor errors or produces mostly correct results.	Code runs with significant errors or produces partially correct results.	Code does not run or fails to produce correct results.
Code Quality	Code is well-organized, readable, and properly commented.	Code is somewhat organized, with some readability or commenting issues.	Code is poorly organized, with little attention to readability or comments.	Code is disorganized, difficult to read, and lacks comments.
Adherence to Instructions	Followed all instructions and guidelines precisely.	Followed most instructions with minor deviations.	Followed some instructions but with several deviations.	Did not follow instructions or guidelines.

Total Possible Points: 100

- **Completion:** 25 points
- **Functionality:** 40 points
- **Code Quality:** 20 points
- **Adherence to Instructions:** 15 points

Learning Objectives

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

1. **Understand and Apply Core Python Concepts:**
    - Develop a solid understanding of Python syntax, data types, control structures, and functions.
    - Write Python code to solve basic to intermediate programming problems.

2. **Work with Data Structures:**
    - Create and manipulate lists, dictionaries, and other data structures to store and process data effectively.
    - Use Python's built-in data structures to implement algorithms and solve complex problems.

3. **Implement Object-Oriented Programming:**
    - Design and implement classes and objects to model real-world problems.
    - Understand and apply principles of object-oriented programming, including inheritance and polymorphism.

4. **Manage Files and Handle Exceptions:**
    - Read from and write to files using Python, managing file input/output efficiently.
    - Implement error handling using try-except blocks to create robust and fault-tolerant programs.

5. **Develop Interactive Programs:**
    - Create programs that interact with users through input and output, including text-based and graphical interfaces.
    - Build interactive applications, culminating in the development of a simple game using Pygame.

6. **Work with External Data and APIs:**
    - Fetch and process data from external sources using APIs.
    - Implement programs that download, parse, and visualize data from the web.

7. **Collaborate on Software Projects:**
    - Use GitHub for version control, collaboration, and sharing code with peers.
    - Participate in collaborative coding projects, including code reviews and team-based development.

8. **Complete a Comprehensive Final Project:**
    - Apply the skills learned throughout the course to develop a complete game (Alien Invasion) as a final project.
    - Demonstrate the ability to plan, code, debug, and present a fully functional software application.

9. **Engage in Problem Solving and Creativity:**
    - Approach programming challenges with creativity and critical thinking.

- Develop solutions to open-ended problems through projects and coding exercises.

10. Reflect on Learning and Growth:

- Assess personal progress in learning Python and programming concepts.
- Identify areas for further development and set goals for continued learning in computer science.

Mental Health Matters:

The Gable Health and Counseling Center offers students the chance to meet with therapists at no charge. Students are encouraged to make appointments to receive confidential care for small and large issues. If you, or anyone close to you on campus, are suffering from any mental health issues, you are encouraged to reach out and use the services on campus to get the care you need. The office is open from Monday through Friday 8:30am - 4:30pm and appointments are scheduled from 10-6 M-TH and 9-5 on Fridays. Students can set up a meeting with a therapist on campus by walking to the Gable Health Center located on campus at 1829 Linden Street or by calling the Gable Health Center at 610-921-7532.

Academic Integrity:

Integrity provides the foundational trust upon which academic communities are built. The Albright College academic community subscribes to an Academic Integrity Code that expresses this vital importance of academic honesty and integrity. By accepting membership in this community, we each assume the obligation to be trustworthy in all pursuits. Cheating, plagiarism, collusion, dishonesty, tampering with or damaging computers or other property, or any other form of academic misconduct are unacceptable and may result in sanctions up to dismissal from the class or the College.
In-class exams or quizzes must be completed by you without assistance from any person or source. Assignments not designated as paired or group projects must be your own work, and paired or group projects must be the work product of the group. For individual or group assignments (labs, projects, homeworks, etc.) the “empty hands policy” will prevail: You may freely discuss ideas with other students, but each student or group must leave the discussion without written, photographed, or otherwise recorded material. This includes sending, copying, or transcribing algorithms, solutions, or programming code from any local or online source into your own work. Any manifestation of submitting another person’s

work for your own (digital, verbal, written, etc.) is an academic integrity code violation that is not permitted under this policy and will be treated accordingly.