CS118: Fundamentals of Computer Programming

Instructor: **Matthew Kindy**
Email: **Matthew.Kindy@ERAU.edu**
Office: **LB116**
Office Hours: **See Canvas and office door**
Course Credits: **3 Credit Hours**

Course Description
This is an introductory course in computer programming using Python programming language. This course uses a problem-solving approach for developing algorithms. Algorithms will be implemented in Python and include the following topics: data types and related operations, looping, decision, input/output, simple data structures such as arrays and tuples, functions, and files.

Pre-requisites: None Co-requisites: None.

Course Goals
The course should provide the student with a tool for solving problems. Proper programming practices should be emphasized. The student should be able to develop modular, structured-programming solutions to reasonably sophisticated problems by the end of the semester. After successful completion of this course the student should be able to:

- Develop algorithmic solutions to various problems.
- Implement the solutions (algorithms) using the Python language.
- Employ testing techniques to verify the correctness of the solutions.

Learning Objectives

- Create, edit, and execute Python programs.
- Understand the role of algorithmic design as it applies to solving problems using computers.
- Design and implement algorithmic solutions to problems requiring elementary processing concepts of arithmetic, basic data types and operations.
- Design and implement algorithmic solutions to problems requiring the basic control structures of sequences, selection, and repetition
- Design and implement algorithmic solutions to problems requiring array structures,
- Design and implement algorithmic solutions to problems requiring the application of linear search and sorting.
- Design and implement solutions of intermediate complexity requiring the use of I/O with files.
- Design and implement solutions of intermediate complexity using functions.
- Design and implement solutions of intermediate complexity requiring the use of non-numerical data such as Booleans, characters, and strings.

Textbook
None required.

Textbook Reference Options

Think Python - How to Think Like a Computer Scientist
Author: Allen B. Downey
ISBN-13: 978-1491939369
Web version: http://www.greenteapress.com/thinkpython/html/index.html
PDF Download: http://www.greenteapress.com/thinkpython/thinkpython.pdf
Hard Copy purchase: https://www.amazon.com/gp/product/1491939362

Starting Out with Python plus MyProgrammingLab with Pearson eText – Access Card Package 3/e or 4/e
Author: Tony Gaddis
MyLab + Student Value Edition with etext
3/e ISBN: 9780133862263
4/e ISBN: 9780134444321

Supplemental Material – Python 3

Windows: Use the installation files on Canvas
Their Sources: Python: https://www.python.org/downloads/windows/
Spyder3: https://www.spyder-ide.org

OS X: Use the installation files on Canvas
Their Sources: Python: https://www.python.org/downloads/mac-osx/
Spyder: https://www.spyder-ide.org

Grading
Your course grade will be determined using the following weights:

Exercise sets* (approx. 30)	20%
Labs (approx. 10)	20%
Programs (approx. 10)	20%
Exams (4)	40%

** Exercise grade is assigned 20 percentage points for achieving an overall average of 75.00% or more; 15 percentage points for achieving an average between 50.00% and 74.99%; and 8 percentage points for achieving an average between 40.00% and 49.99%.*

The cutoffs for final grades are:

A	B	C	D	F
>= 90.00%	89.99% – 80.00%	79.99% – 70.00%	69.99% – 60.00%	59.99% – 0.00%

Extra credit opportunities may arise as the professor sees fit and beneficial for students. This course does not have a predefined breakdown of letter grades awarded – you get the grade you earn, so hopefully all students will earn an A!

Late Assignment Policy
Unexcused: All links for assignment submissions will disappear at the exact time of the due date. No late homework, in-class activities, modeling problems, or final projects will be accepted. Meaning all work not submitted (or submitted late via email) will be a zero (0). ***Please plan accordingly to ensure your homework is graded.*** One idea to plan accordingly is to submit your homework at least an hour early and then resubmit your homework once complete, if you are continuing to work on it right up until the due date. A better idea is to start your homework early and instead of working on it until the last minute it is due.

Excused (note from the dean of students or appropriate oversight body – example: ROTC, Athletics): The only exception to this will be an excused absence from the dean of students for all days that the assignment was given. Premeditated excused absences (e.g. athletic trips, planned events) require notice at least 24 hours before the first missed class day.

Missed Exam Policy
Unexcused: If you miss an exam and do not have an excused absence, email your professor within 24 hours for information on how to proceed. Failure to reach out to your professor within 24 hours will result in an automatic zero.

Excused: (Requires note from the dean of students or appropriate oversight body) If you miss an exam with an excused absence, you may choose to either take the score of the next exam for the both that next exam and the exam you missed; or you may schedule a make-up exam at the professor's convenience. Premeditated excused absences (e.g. athletic trips, planned events) require notice at least 24 hours before the first missed class day.

Reassessment Policy
One role of instructors is to determine what is pertinent in the grading scheme for their courses. Any and all assessment of a grade and/or method of determination is at the discretion of each instructor. If a student believes an error has been made in the scoring of an assignment, quiz, exam, project, or final grade in the course, the student has seven calendar days from the date of grade posting to request from the instructor (NOT the grader) a review of the scoring. This request must be via email to the instructor. The instructor is free to reassess items as the instructor feels appropriate, including regrading entirely without consideration of previous determinations. The instructor is free to choose a third party (such as a grader) to review the item, but is not obligated to accept that third party's assessment.

Professionalism
Submitting work: If a submission for any kind of assignment/project is flagged as inappropriate, profane, disrespectful, harassing, sexist, racist, or offensive, you will receive a zero. If you are unsure if your work is considered offensive, please ask for guidance.

Communication and interactions: When communicating and working with others (peers, faculty, staff, etc.) in person or via electronic communication (emails, texts, etc.), you are responsible for being professional. If a peer has offended you, please notify your professor so the situation can be addressed. Any students that use offensive language or mistreat others will be asked to attend a one-on-one meeting with the faculty and/or depending on the frequency and severity other appropriate measures will be taken to address the behavior.

ERAU Academic Integrity
This course follows the ERAU Academic Integrity Honor Statement. If you are found to be in violation of the policy with regards to any assignment, the consequences will be significant (possibly including failure of the course) and including filing an Academic Integrity Violation Form with the Office for the Dean of Students. (Find out more information at: http://daytonabeach.erau.edu/campus-life/dean-of-students/honor-code/index.html).

If you are found responsible for violating the ERAU Academic Integrity Honor Statement on any task worth equal to or greater than 10% of your course grade (i.e. the final project or any of the exams), you will automatically receive an F in the course.

I encourage you to collaborate with your peers, but I **do not** acknowledge rewriting another student's completed work, sharing lines of code, re-typing what another has written, etc. as acceptable collaboration. I recommend sticking to discussing concepts and logic and **not** reviewing lines of code with your peers. If you are stuck and struggling with problems in your code and discussions about concepts are not helping, I recommend you talk to myself, another professor, or a tutor. (Struggling with a new language or new concepts is not unusual even for experienced programmers – be sure you start early and remember the Late Assignment policy above.) Make sure to credit all peers that you collaborate with, including any tutors or friends that you receive help from on any assignment.