Southern Arkansas University
Course Syllabus

Course Information

Course Number and Title: CSCI 2113/2111 Computer Science II and Laboratory (Programming with Python)
College and Department: College of Science and Engineering, Department of Mathematics and Computer Science
Term: Spring 2023
Format: Face-to-Face (0-24% online)
Course day(s) and time: Monday/Wednesday/Friday 9:00-9:50 AM, 11:00-11:50 AM
Lab: Wednesday/Thursday 1:10-3:00 PM
Course Location: WIL 123
Lab: WIL 113
Weeks in length: 16
Class meetings per term: 15
Hours per class meeting: 3 CSCI 2113, 2 CSCI 2111
Credit hours awarded: 3 CSCI 2113, 1 CSCI 2111

Instructor Information

Instructor: Dr. Islam Akef Ebeid
Contact Number: 870 235 4952
E-mail: iaebeid@saumag.edu
Office Hours: Mondays, Wednesdays, Fridays 3:00 – 5:00 PM

Office: WIL 123A

Course Description

This course is a hands-on experience of various topics in programming using the Python programming language. That includes and is not limited to expressions and variables, input/output, control flow (conditionals and loops), arrays/vectors, strings, functions, pointers, structures, and files. Also, more advanced topics such as object-oriented programming; classes and objects, data encapsulation, inheritance, and polymorphism will be discussed.

Related courses: CSCI 2103. Computer Science I, CSCI 2101. Computer Science I Lab

Course Objectives and Learning Outcomes

The goal of this course is to teach students programming skills that are necessary for them to learn to succeed in different STEM-related careers. In addition, programming is fundamental in industry and the academy. It is fundamental in Business, Liberal Arts, and Social Sciences.

In this course, Python was chosen as a programming language because it is highly demanded in the job market. Moreover, Python provides a very flexible platform that allows both functional and object-oriented programing paradigms. The syntax is easy and flexible and helps the student understand basic programming concepts in the least amount of time. That said it is important to understand that Python is just a vessel and a tool used to teach students how to think like a programmer, or as the chosen book mentions, as a computer scientist. The core skill in programming is logic. That is really what we are teaching the student; how to think logically about a problem that requires computation as a resource. Due to the nature of programming, the topics tend to get complex quickly. Hence, I employ a step-by-step approach where I don't progress with students to the next step until I am sure that the prerequisite skills were mastered.

Here are the core learning outcomes from this course:

1- Refreshing basic concepts in computing like what is a processor and what is memory and what is the most common computer architectures.
2- Acquiring basic concepts in programming that might have not been covered in the prerequisites, such as expressions, variables, branching, input/output, and control flow.
3- Understanding the skill of efficient programming and how to think of the computer as a resource that needs to be used wisely.
4- Digesting more complex topics such as arrays, functions, pointers, and memory management.
5- Object-oriented programming, inheritance, and polymorphism and how they relate to data modeling.
6- An overview of multiple topics in Computer Science as an impactful field in the industry and the academy.
7- The student shall come out of the class with the skills needed to advance to more upper-level courses and topics.
8- In addition, the student will be able to build simple applications using Python. And will be prepared for any basic interview questions about the job market.

Course Format

This course is accompanied by a lab which means that the learning will happen while the students are engaged on a computer in the classroom. In the first half of the class duration, the instructor will introduce the topic. Following the instructor will discuss the assigned readings with the students after the students answer questions on Blackboard regarding the readings. The reading quizzes will consist of 5 questions from the assigned readings. Following the instructor will direct the students to perform the designed in-class assignment. The designed

in-class assignment helps the student understand the topic being taught. The designed in-class assignment will be chosen to take about 2 hours to complete with the option that the student can complete it at home if they couldn't finish in class but that will affect their final grade. The first week will be an introduction. In addition, the students will learn about how to set up their environments.

Various tools will be used to work on the in-class assignment. Some assignments will require a local development environment, some will require educational and commercial platforms such as HackerRank. HackerRank and LeetCode are both the primary programming platforms used in many programming interviews in the job market. HackerRank also provides a university-based platform specifically designed to teach programming in the classroom. Setting up HackerRank only requires students to sign up using their emails. In addition, Google Colab will also be used. Google Colab is a free online Python interpreter and development environment. Setting up Google Colab only requires students to have a free Google account.

After each student finishes the in-class assignment the instructor will ask the students to download their source code files and submit them to Blackboard. The instructor will then discuss the solution to the previous week in class assignment. The current in-class assignment will then be graded separately. Note that the student's solution doesn't have to match the answer key, nor does it have to be 100% correct. The student's solution should show effort and the logic used behind their code explained in comments. The in-class assignment shall take approximately 2 hours with the option of the student taking the assignment home. During that time, the instructor shall assist the students while completing the in-class assignment.

Starting weeks (2-15) except for week 8 we will roughly follow the required textbook. Week 8 will be dedicated to the in-class midterm evaluation exam. While finals week will be dedicated to an in-class final exam. Both the midterm and the final exam will consist of two parts. The first part will be multiple choice questions and the second part will be a programming task.

Course Material

The required textbook is:

Think Python 2: How to think like a computer scientist, available legally for free using the following link:

https://open.umn.edu/opentextbooks/textbooks/think-python-how-to-think-like-a-computer-scientist

https://greenteapress.com/wp/think-python-2e/

The book is free because it is offered on an Open Education Resources database under the CC BY-NC license.

There will also be slides for each class. The slides will be uploaded regularly to Blackboard.

The in-class programming assignments, the midterm, and the final exam will come from the required textbook and lectures.

In-class assignments will be inspired by tasks available on:

https://leetcode.com/

https://www.hackerrank.com/

https://github.com/AllenDowney/ThinkPython2

In addition to the exercise available in the book after each chapter.

Grade Structure

Criteria	Number of Occurrences	Points per Occurrence	Total Points	Percentage of Total
Attendance	16/16	8	128	12.8%
Participation	16/16	2	32	3.2%
Wednesday Reading Quiz	12/16	10	120	12%
Friday Book Exercises	12/16	10	120	12%
Lab Inclass Assignments	12/16	25	300	30%
Midterm Exam Part 1	1/16	75	75	7.5%
Midterm Exam Part 2	1/16	75	75	7.5%
Final Exam Part 1	1/16	75	75	7.5%
Final Exam Part 2	1/16	75	75	7.5%
Total	16/16	-	1000	100%

- Attendance is 1 point for each class (Monday, Wednesday, and Friday + Lab) so the total is 4 points.
- Participation throughout the week is 1 point (this means that the instructor expects the student to interact at least once by asking a question)
- Midterms and Finals will be half multiple-choice questions from lectures and books and half a programming task.
- The in-class assignment will be comprised of 1 to 8 tasks depending on the week. If the class ran out of time the remaining tasks would be a take home assignment.
- On Friday's points will be working on exercises inspired by the book and they will be worth 2 points.

Grade Policy

A+	95%-100%
A	80%-95%
B	70%-80%
C	60%-70%
D	50%-60%
F	Below 50%

Please note that to get a full grade on the in-class programming assignment or task you need to provide a solution to the programming problem. Incorrect outputs will be considered if a sufficient effort was shown and demonstrated. A+ scores will be reserved for extraordinary answers.

Note that the student can convert the class to an honors course by finishing a project guided by the instructor. Please reach out to the instructor directly if this is your case. The honors student will work on a final project separate from the class and will provide a report, and presentation to the instructor during the final's week.

Course Plan

Week	Monday	Wednesday	Friday	Lab (Wednesday/Thursday)
1 (January 11)	• Introduction	• Introduction	• Introduction	• Syllabus
2 (January 18)	• Processors, Operating Systems, Compilers, and Programming Languages	• Chapters 1,2: Variables, Expressions, and Statements • Reading Quiz	• Chapters 1,2: Exercises from the book	• Hello world task 1 (local environment) • Hello world task 2 (local environment) • Hello world task 3 (local environment) • Arithmetic operators (local environment)
3 (January 25)	• Memory, Procedures, and Functions	• Chapter 3: Functions • Reading Quiz	• Chapter 3: Exercises from the book	• If-Else (local environment) • Print Function (local environment) • Setup HackerRank • Calendar Task (HackerRank)
4 (February 1)	• Booleans, Recursion, The Stack, and The Heap	• Chapter 5: Conditionals and recursion • Reading Quiz	• Chapter 5: Exercises from the book	• Setup Google Colab • Loops task 1 (Google Colab) • Loops task 2 (Google Colab) • List Comprehensions (Google Colab)
5 (February 8)	• Data Structures and Algorithms	• Chapter 6: Fruitful functions • Reading Quiz	• Chapter 6: Exercises from the book	• Find the Runner-Up Score (HackerRank)

				• Finding the Percentage (HackerRank)
6 (February 15)	• Data Modeling, Abstraction, Ontology, Database, and Object Orientation	• Chapter 7: Iterations • Reading Quiz	• Chapter 7: Exercises from the book	• Nested Lists (local environment) • Lists (local environment) • Numpy Arrays (local environment) • SciPy Matrices (local environment)
7 (February 22)	• Natural Language Processing and Understanding: Working with Text	• Chapter 8: Strings • Reading Quiz	• Chapter 8: Exercises from the book	• String Split and Join (local environment) • String Mutations (local environment) • Tokenization (Google Colab) • Lemmatization (Google Colab)
8 (March 1)	• Midterm Course Recap	• Midterm Part 1A	• Midterm Part 1B	• Midterm Part 2
9 (March 8)	• Computer Vision: Working with Images	• Chapters 10, 11, 12: Tuples, Sets, Dictionaries, and Hash Tables • Reading Quiz	• Chapters 10, 11, 12: Exercises from the book	• Chapter 4: Case Study with Google Colab
10 (March 15)	• Objects, Serialization, and Object-Oriented Design	• Chapter 14: Files • Reading Quiz	• Chapter 14: Exercises from the book	• Chapter 13: Case Study with Google Colab
11 (March 22)	Spring Break	Spring Break	Spring Break	Spring Break
12 (March 29)	• Classes and Inheritance	• Chapter 15, 16: Classes,	• Chapters 15, 16: Exercises from the book	• Python OOP task 1 (HackerRank)

		Objects, and Functions • Reading Quiz		• Python OOP task 2 (HackerRank)
13 (April 5)	• Encapsulation and Polymorphism	• Chapter 17, 18: Methods and Inheritance • Reading Quiz	• Chapter 17, 18: Exercises from the book	• Python OOP task 3 (local environment) • Python OOP task 4 (local environment)
14 (April 12)	• Design Patterns	• Chapter 19: The Goodies • Reading Quiz	• Chapter 19: Exercises from the book	• HackerRank Contest 1
15 (April 19)	• Machine Learning in Python	• Appendix A, B: Exceptions, Debugging, Error Handling and Analysis of Algorithms • Reading Quiz	• Appendix A, B: Exercises from the book	• HackerRank Contest 2
16 (April 26)	• Ethics in Computer Science	• Course Recap	• Course Recap	• Course Recap
17 (May 1)	Finals Week	Finals Week	Finals Week	Finals Week

University Policies
For general university policies please refer to: https://web.saumag.edu/academics

Academic Dishonesty
Southern Arkansas University affirms its commitment to academic integrity and expects all members of the University community to accept shared responsibility for maintaining academic integrity. Students in this course are subject to the provisions of the University's Academic Integrity Policy, approved by the president and published in the Student Handbook. Penalties for academic misconduct in this course may include a failing grade on an assignment or a failing grade in the course. Continued enrollment in this course affirms a student's acceptance of this University policy.

Code is like writing. It only belongs to its author. The author can refer to somebody else's work, but they must cite them and give them credit for it. When you take someone else's code, you are robbing that person of the time that they spent authoring that piece of code. If you are facing a problem in programming it is ok to go online and search for a solution, but you must mention where you got that piece of code from. The whole point is not to trick people into thinking that you wrote something that you did not. That is your measure. Keep it in mind.

And finally, the goal of this class is to teach you a skill. If you plagiarized, copied, cheated, or fabricated you are hurting yourself before others.

In the context of this class here is a list of what I consider academic dishonesty:

1- While in class you looked at your colleague's screen and copied the same code, she/he has written

2- I might ask you to continue your in-class assignments at home. So, you went home and asked your sibling who is a software engineer to do it for you.

3- You copied code from the internet without citing it

4- During an MCQ exam you copied the whole question and pasted it on google in the hopes of finding the correct answer

Please refer to https://web.saumag.edu/academics/dishonesty-and-integrity/

Disability Support Services
It is the policy of SAU to accommodate students with disabilities, pursuant to federal law, state law, and the University's commitment to equal education opportunities. Any student with a disability who needs accommodation should inform the instructor at the beginning of the course. Students with disabilities are also encouraged to contact the Office of Disability Support Services, Reynolds Center, Room 216, 870-235-4145. More information can also be found at https://web.saumag.edu/testing/disability-support-services/.

Instructor Policies

Holidays

The instructor will follow the federally and state-recognized holiday schedule by the University which can be found here:

https://web.saumag.edu/human-resources/holiday-schedule/

However, if you need special accommodation for religious or other types of holidays that you observe please let the instructor know beforehand.

Diversity

The instructor is committed to diversity, inclusion, and equality in the classroom and accordance with the university policies regardless of any cultural background, country of origin, religion, race, ethnicity, and sexual orientation.

Please let the instructor know in what way you would like to be addressed. The instructor will ask the students during the first lecture about their names, pronouns, and other forms of addressing they would like to be referred to. If that changed or if you were addressed mistakenly, please notify the instructor.

Office Hours

- First come, first served.
- Maximum 15 minutes if people are waiting,
- If 2 or more students come at the same time will be in the order of the last name
- Please come prepared to office hours with questions.

Conflict

In case of conflict between students in the classroom, the instructor will act as a mediator until proper university authorities are notified.

How to succeed in this class

If you worked hard, put in the effort, and used all your resources to achieve the highest grade in this class you will receive it. In that regard I recommend the following to succeed in this class:

1- Please attend all classes and let the instructor know if you will not be able to attend.
2- Please work separately on your own during the in-class and homework assignments unless otherwise made clear.
3- Using internet resources is allowed with restrictions that the instructor will mention during the classroom. Yet please don't copy and paste code or answers for any questions.
4- If you are having trouble finishing an in-class assignment that will NOT automatically result in a bad grade. If you showed effort and provided explanations of your thought process a good grade could be achieved.

5- Complete all your assignments to the best of your abilities.
6- See your errors and mistakes as opportunities to learn more.
7- Please ask questions if things are not clear.
8- Use email as the preferred way of communicating.

Academic Resources

Mental Health

If you need help with any issue that is affecting your academic performance, please refer to:

https://web.saumag.edu/counseling/

Writing

If you have difficulty communicating in written English language, please let the instructor know and please refer to:

https://web.saumag.edu/writing-center/

Tutoring

If you would like additional help for the class or any other classes, please notify the instructor and refer to:

https://web.saumag.edu/support/tutoring/