Course Title: CS 126 Computer Science I
Term: Winter 2023
Instructor: TBA
Course Credit: 3
Mode of Instruction: Online

Course Description:

This course introduces students to the foundational principles of computer science, including object-oriented fundamentals. It includes an overview of the various tools available for writing and running Python, and gets students coding quickly by studying algorithm design, testing and debugging, top-down design and reading and writing to files. Topics also cover data types, control flow, object-oriented programming, and graphical user interface-driven applications. The objective is to help students use the Python programming language to solve problems.

Course Prerequisites:

MAT 125 Precalculus Mathematics

Learning Outcomes:

By the end of the course, the student should be able to:

A. Develop understanding of computer science, data structure and algorithm;

B. Learn a language for expressing computations—Python;

C. Design and analyze algorithms to identify, formulate, and solve computational problems;

D. Master the process of writing and debugging a program;

E. Write programs to solve real-world problems, and come away with everything you need to produce quality code.

Course Material:

Paul Gries, Jennifer Campbell, Jason Montojo, Tammy Coron, *Practical Programming: An Introduction to Computer Science Using Python 3.6*, 3rd Edition, 2017.

Evaluation:

- 2 Projects [20%]

- 2 Assignment [20%]

- Mid-term Exam [25%]

- Final Exam [35%]

Description of the Evaluation tasks:

Assignment/ Essay/ ... : During the term, students will be required to finish several evaluation tasks within due date. All the tasks are linked with specific course topics/outcomes and will adequately assess students' competence and learning outcomes. Students are encouraged to meet with instructor about these tasks at any point.

Mid-term/ Final Exams/ Quiz/... : There may be periodic quizzes given at the beginning of lecture sessions; the feedback from these quizzes will monitor the progress of the learners and help to set learning priorities. There will be mid-term exam/ final exam for the course. They are the basic criteria for the evaluation of students' learning outcomes and final grade.

Grading Policy:

Students are supposed to finish each online lecture. Prior to each class, students should finish the required readings. During the class time, students are encouraged to make use of all relevant online course resources and communicate with the instructor. Students' grades are accumulated based on the cumulative evaluations.

Students' letter grade will be assigned according to the following scale:

A+ 90-100	A 85-89	A- 80-84
B+ 77-79	B 73-76	B- 70-72
C+ 67-69	C 63-66	C- 60-62
D+ 57-59	D 53-56	D- 50-52
F < 50		

Academic Integrity:

Students must strictly adhere to the university's academic integrity rule; and all essays, exams and any other form of academic assignments must adhere to these rules. Any form of plagiarism, cheating, or misappropriation of materials will be considered a violation of academic integrity and will be punishable by the university.

Withdrawal from the Course(s):

Students will be able to apply for a transfer or withdrawal within 3 days of the starting date of the course. If a withdrawal is applied for within 3 working days, the tuition fee will be fully refunded. After 3 days, the tuition fee will not be refunded. If a withdrawal is applied for in the first two weeks, it will be recorded as W (Withdraw) on the course transcript. After this initial two-week period, the class will be recorded as F (Fail).

Tentative Schedule:

1	What's Programming?
2	Hello, Python: How Does a Computer Run a Python Program?
3	Describing Code Making Code Readable
4	Designing and Using Functions
5	Writing and Running a Program **Assignment 1**

6	Working with Text
7	Making Choices
8	A Modular Approach to Program Organization
9	Using Methods: Modules, Classes, and Methods
10	Storing Collections of Data Using Lists **Project 1**
11	**Midterm Test**
12	Repeating Code Using Loops
13	Reading and Writing Files
14	Writing Algorithms That Use the File-Reading Techniques
15	Storing Data Using Other Collection Types **Assignment 2**
16	Designing Algorithms
17	Top-down Design
18	Searching and Sorting
19	Object-Oriented Programming
20	Testing and Debugging **Project 2**
21	Creating Graphical User Interfaces
22	Expressions and Values: Arithmetic in Python
23	Databases
24	Final Exam Reviews
25	**Final Exam**