

Course Title: CS 126 Computer Science I

Term: Summer 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. Students will learn the core principles of object-oriented programming and how to apply them to solve real-world problems. Key topics such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction will be covered. Through hands-on programming exercises, students will gain practical experience in designing and implementing object-oriented programs using a specific programming language.

Course Prerequisites:

MAT 125 Precalculus Mathematics; CS 110 Computer Science Principles

Learning Outcomes:

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

- A. Develop understanding the fundamental principles and concepts of computer science.
- B. Design and analyze algorithms to identify, formulate, and solve computational problems.
- C. Master the process of writing and debugging a programming.
- D. Collaborate effectively in a team to develop object-oriented programs.
- E. Demonstrate good software engineering practices, including code documentation and testing.

F. Analyze and solve problems using object-oriented techniques.

Course Material:

Principles of Computer Science, Salem Press, Salem Pr, Year: 2016.

Object-Oriented Programming in C++, Robert Lafore, Sams Publishing, 2005.

Evaluation:

- 2 Programming Projects [30%]
- Project Report [15%]
- Mid-term Exam [20%]
- 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	Introduction to Object-Oriented Programming (OOP)
2	Overview of computer science and programming paradigms

3	Biometrics
4	Cloud Computing
5	Computer Animation Programming Project 1
6	Classes and Objects
7	Constructors and Destructors
8	Inheritance and Polymorphism
9	Encapsulation and Information Hiding
10	Getters and Setters
11	Midterm Test
12	Abstraction and Interface Design
13	Abstract Data Types
14	Object-Oriented Analysis and Design
15	Introduction to Unified Modeling Language
16	Use cases, Class Diagrams, and Sequence Diagrams Programming Project 2
17	Applying OOP Principles in System Design
18	Exception Handling
19	Handling Runtime Errors Using Exceptions
20	File I/O and Serialization
21	Object Serialization and Deserialization
22	Generics and Templates Project Report
23	Software Testing and Debugging
24	Software Engineering Best Practices
25	Final Exam