TROY UNIVERSITY
CS 256
Computer Science 2

COURSE SYLLABUS

Spring 2025

PRE-REQUISITES:
CS 255- Computer Science 1

INSTRUCTOR INFORMATION:
Dinh-Van nguyen, Ph.D
Dept. of Communication Engineering, SEEE, Hanoi University of Science and Technology
Troy Faculty Member
Email: van.nguyendinh@hust.edu.vn

INSTRUCTOR EDUCATION:
Ph.D. in Computer Science, 2018, Mines ParisTech University, France
M.S. in Computer Science, 2015, Hanoi University of Science and Technology, Vietnam
B.S., in Computer Science, 2012, Korea Advanced Institute of Science and Technology

CATALOG DESCRIPTION:
A continuation of the introduction to the theory and development aspects of a high level programming language. The course provides students with a foundation in problem solving, algorithm development, and program design using C++. It covers enumeration, repetition, multi-dimensional array and string manipulation, structures, advanced files input/output and an introductory to class concept in C++. Program analysis, design, development and testing are emphasized

STUDENT OUTCOMES:
- Understand Class concept in C++
- Implement solutions in object oriented programming paradigm
- Understand advanced structures such as linked lists, queue, stacks.
- Implement advanced files operation
- Analyze, develop solution and debugging problem in C++

Students will demonstrate the above through their performance on the assigned programming projects, exams and discussion Black Board exercises.

GRADING
Attendance – 10%
Individual programming projects - 30%
Team programming Project - 30%
Final Exam - 30%

GRADING SCALE:
Grades will be assigned according to the following scale:

A	90 - 100
B	80 - 89
C	70 – 79
D	60 – 69
F	below 60

TEXTBOOK
Tony Gaddis, Starting out with C++, Six Edition, Pearson, 2009, ISBN: 978-0321545886
A C++ compiler is needed. Microsoft visual studio or C++ express are recommended for the course. You can download them free online as the instructions provided. "Computer problems" are not an acceptable excuse if you cannot turn in homework by the deadlines.

METHODS OF INSTRUCTION: In-Classroom Lectures, Labs, Tests, Handouts/Homework, Class Participation and Reading assignments.

HONESTY AND PLAGIARISM

*****Plagiarism is defined as submitting anything for credit in one course that has already been submitted for credit in another course, or copying any part of someone else's intellectual work – their ideas and/or words – published or unpublished, including that of other students, and portraying it as one's ownAll students are required to read the material presented at: http://troy.troy.edu/writingcenter/research.html

All material submitted for grade must be the student's own work.

Anyone found cheating and/or copying will receive an automatic 0 for that assignment or exam or dismissal from the course. This goes for the person who copies as well as the person who allows their work to be copied. Serious penalty (e.g: one lower letter grade) will be given for cheating and plagiarism and students will be required to retake a course if they get D or worse for that course.

OTHER POLICY
There will be no make up test. A missed test or exam will result in 0 points. Contact me in advance in case of emergency such as illness. An original letter address to me on a letterhead paper from a physician or hospital stating that you could not take the test or exam as scheduled is necessary for me to consider your case.

PROJECTS

Students are required to submit their source code file(s) as email attachments prior to the project deadline. In the event you are unable to complete a project or make it work correctly, be sure to email your latest source code to receive partial credit. Students failing to submit projects within the allotted time will receive a project grade of zero and will not be allowed to make late submissions. Exceptions will be made ONLY under extenuating circumstances and ONLY with prior approval by the instructor. To receive full credit, projects must be submitted on or before the due date.

TENTATIVE SCHEDULE

Lecture 1	Course Introduction	Gaddis, Chapter 1
Lecture 2	Pointers and Functions (revise)	Gaddis, Chapter 6,7,8,9
Lecture 3	Introduction to Class (revise)	Gaddis, Chapter 13
Lecture 4	Individual programming project	
Lecture 5	More about class	Gaddis, Chapter 14
Lecture 6	Inheritance, Polymorphism and Virtual functions	Gaddis, Chapter 15
Lecture 7	Inheritance, Polymorphism and Virtual functions	Gaddis, Chapter 15
Lecture 8	Team project idea presentation	
Lecture 9	Individual programming project	
Lecture 10	Debugging	
Lecture 11	Recursion	Gaddis, Chapter 19
Lecture 12	Linked List, Stack & Queue	
Lecture 13	Individual programming project	
Lecture 14	Team project evaluation	
Lecture 15	Final Exam	