CSCE 1337.0w2 Course Syllabus

Course Information

Course Number/Section	CSCE 1337.0W2
Course Title	Computer Science 1
Term	2025 Spring
Classroom Days/Time	Mon, Wed 11:30 AM – 12:45 PM
Classroom	Online Blackboard Collaborate

Professor Contact Information

Professor	Scott Dollinger
Office Phone	214-430-0036
Email Address	Scott.Dollinger@utdallas.edu
Office Location	Contact by Phone During Office Hours
Office Hours	Mon, Wed 1:00 PM – 02:00 PM

All e-mails must have the **Course.section** in the e-mail subject title, or the **e-mail will not get a response**.

If you have any special problems, p*lease communicate to me via e-mail* as soon as reasonably possible.

TA/GRADER CONTACT INFORMATION

To Be Announced, when assigned will be posted to e-learning under **Contact Help Information** Page in Blackboard Navigation Pane.

COURSE PREREQUISITES:

Prerequisite: CS 1436 with a grade of C or better or equivalent.

Course Modality

Instructional Mode

This is an online class session course that you must attend.
Class sessions are online using Blackboard Collaborate to run the online Blackboard virtual class sessions.
For any assistance you may need with Blackboard and/or Blackboard Collaborate, contact the eLearning Help Desk..
There are not any lecture or class recordings.
Class material will be posted to blackboard course page as the class progresses

Course Platform/Mode

This course uses Blackboard.
See the following guides for information:

https://dox.utdallas.edu/manual1073
https://ets.utdallas.edu/elearning/resources/software/blackboard-ultra

Course Attendance

The student is expected to attend all classroom class sessions.
Students who fail to attend class regularly are inviting scholastic difficulty. Successful class participation is defined as consistently adhering to university requirements, as presented in this syllabus.

Attendance is not a grading component for this course.

Class Participation

The student is expected to read the assigned readings from the book to fully understand the course topics.

COVID-19 Guidelines and Resources

The information contained in the following link lists the University's COVID-19 resources for students and instructors of record. Please see http://go.utdallas.edu/syllabus-policies.

Class Materials

The instructor may provide posted class materials that includes the syllabus, schedule, assignments, slides, and sample demo programs. These materials are intended to supplement the classroom experience.
The materials will be that will be made available to all students registered for this class on the Blackboard system.
These materials may be downloaded during the course; however, these materials are for registered students' use only.

Classroom materials may not be reproduced or shared with those not in class or uploaded to other online environments except to implement an approved AccessAbility Resource Center accommodation.

Cell phones usage is *not* allowed in the classroom. You may not audio record, take pictures or videos in the classroom.
The course in classroom activities and lectures are the copyrighted material of the University of Texas at Dallas. University of Texas at Dallas will prosecute any violations against copyright violations.

Failure to comply with these University requirements is a violation of the Student Code of Conduct.

Course Description

Review of control structures and data types with emphasis on structured data types.
Applies the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design.
Includes basic analysis of algorithms, searching and sorting techniques, and an introduction to software engineering concepts.
The programming language used in the course is C/C++.

This course is the middle one in the UTDCS programming sequence. Goal is to build your coding expertise in these courses so that you can apply your skills to complete assignments/projects in all the future courses, without much handholding.

CS 1436 Programming Fundamentals (C++)
CS 1337 Computer Science I (C++)
CS 2336 Computer Science II (Java)

Student Learning Objectives/Outcomes

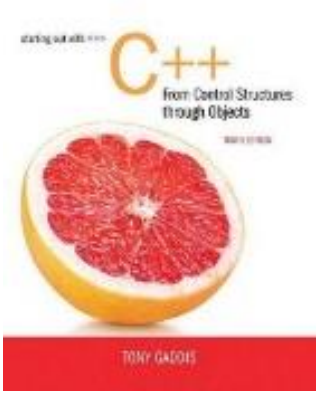
After successful completion of this course, the student should have an:

1. Ability to implement linear and binary searches.
2. Ability to implement simple sorting algorithms.
3. Ability to implement structured data types.
4. Ability to define and implement a class.
5. Ability to use fundamentals of object-oriented design.

Required Textbooks and Materials

Required Texts

Textbook:



Starting Out with C++ from Control Structures to Objects, 9th Ed
Tony Gaddis
ISBN-13: 97801344983799
ISBN-10: 01344983720
© Pearson 2017-02-23

Textbooks and some other bookstore materials can be ordered purchased at the UT Dallas Bookstore.

Class Materials

The instructor may provide posted class materials that will be made available to all students registered for this class as these are intended to supplement the classroom experience.

These materials may be downloaded during the course; however, these materials are for registered students' use only.

Classroom materials may not be reproduced or shared with those not in class or uploaded to other online environments except to implement an approved AccessAbility Resource Center accommodation.

Cell phones usage is *not* allowed in the classroom. You may not audio record, take pictures or videos in the classroom. The course in classroom activities and lectures are the copyrighted material of the University of Texas at Dallas. University of Texas at Dallas will prosecute any violations against copyright violations.

Failure to comply with these University requirements is a violation of the Student Code of Conduct.

C/C++ Development Tools

You can use whatever development environment you wish to develop your assignments. It is recommended that you use an Integrated Development Environment (IDE) with source code debugging capability to make your development efforts efficient. The C++ compiler that you use in your development environment must be a C++ 11 version capable compiler. The Visual Studio Community 2022 is used to grade the submissions. If you use a non-Visual Studio IDE, make sure you build and run it in the target platform Visual Studio Community 2022 which is available in THE ECSS computer labs.

You must understand your IDE so that you know how to extract and identify all files (.cpp, .h, input text files … etc.) required for uploading a grading submission. You must submit assignments by zipping up each of the required files in a zip folder. For more information, see the Assignments in the eLearning system home page, in this page look at Submittal and Development Policy.

UTD has an academic license for the Microsoft Azure Academic site, so students may download a free version of Visual Studio 2022 Community and install it on a personal system. Information will be posted to Blackboard Home Page Navigation Menu on how to download a free version of Visual Studio 2022 Community.

Technical Requirements

In addition to a confident level of computer and Internet literacy, certain minimum technical requirements must be met to enable a successful learning experience.
Please review the important technical requirements on the UTD Getting Started with eLearning webpage.

Course Access and Navigation

This course can be accessed using your UT Dallas NetID account and password on the eLearning (Blackboard) course website.
For all students that are properly registered in the course, UTD will automatically install the student's NetID account logon to the eLearning course roster.
Please see the course access and navigation section of the Getting Started with eLearning webpage for more information.
To become familiar with the eLearning tool, please see the Student eLearning Tutorials webpage.

UT Dallas provides eLearning technical support 24 hours a day, 7 days a week.
The eLearning Support Center includes a toll-free telephone number for immediate assistance: 1-866-588-3192
The center also provides an Email request service, and an online chat service.

Communication

This course utilizes online tools for interaction and communication. Some external communication tools such as regular email and a web conferencing tool may also be used during the semester.

Student emails will be answered within 3 working days under normal circumstances.

All e-mails must have **Course.section** in the e-mail subject title, or the **e-mail will not get a response**.

Server Unavailability or Other Technical Difficulties

The University is committed to providing a reliable learning management system to all users.

However, in the event of any unexpected server outage or any unusual technical difficulty which prevents students from completing a time sensitive assessment activity, the instructor will provide an appropriate accommodation based on the situation.
Students should immediately report any problems to the instructor and also contact the online eLearning Help Desk.

The eLearning Help Desk will work with the student to resolve any issues at the earliest possible time.

Grading Policy

Assessment	Percentage
Test 01	20
Test 02	20
Assignments	60

Range	Grade
97 – 100	A+
93 – 96	A
90 – 92	A-
87 – 89	B+
84 – 86	B
80 – 83	B-
77 - 79	C+
73 - 76	C
70 - 72	C-
60 - 69	D
00 - 59	F

Each range shown above is inclusive and without any rounding-up.
For example, 93-96 for grade A is for the score falling in the range between >= 93 and <= 96.

In blackboard, final numeric grades are automatically rounded down to an integer.
A grade of 96.999 is evaluated as a 96, a letter grade of A, not A+.

In eLearning, the Running total in your gradebook shows the current weighted grade, based only on your graded work - what you have submitted and is graded. For example, if you have done only Test 1, Assignment 1, Weekly postings so far (but you have missed Test2 and missed Assignment 2), current total grade will be based on only those entries that you have submitted and done that are graded. We will try to enter 0s for missed work as much as possible, but the student is responsible for understanding the current and updated grades.

Grading Adjustment Regarding Course Average Test Scores

The final numeric grade may be adjusted depending on the final average test scores.
If the final test average is less than 70%, if there is a final average higher assignment score, then regardless of the final average higher assignment score will not be used to calculate the final course score. In such a case, the final course score will be adjusted to a 70%, a C-.

So, if your final average assignment score is 100 %, but your tests average score is 65, the final course score will be a 70, a C-**.**

CS/CE 1337 Required Final Score To Take Future Courses Course Prerequisite for CS 2336/2337:

CS/CE 1337 is a prerequisite for Future Courses in the Computer Science degree track.
A CS/CE 1337 grade of C or better or equivalent is required to proceed to the future courses.

Extra Credit

Make-Up Assignments

Assignments that received a score of less than 75%, can be made up (re-submitted), but will be graded out of a maximum score of 75.

Such make up assignments must be submitted by the due date and time for makeups listed in the course schedule.
During the start of the week that the makeup assignments are due, the submit button will be made available for submitting make-up assignments.

Be careful, Make-up assignments do not have late days.

For more details, see the Assignment Guides in the Assignments in the Navigation Menu in Blackboard.

Late Work

Assignments that are handed in late will be scored as follows:

Days Late	Graded Out Of
1	95
2	90

After 2 late days, assignments will **not** be available for late submittal, but may be submitted later as make-ups.

The last assignment in the course usually does not have any late days. Keep up with and check the course schedule to be aware of such issues.

Grading Criteria

Program Assignments

Historically students that skip programming assignments, or do not put much effort into their programming assignments, or get too much help from classmates, mentors, or others, do not perform well on exam questions testing the material covered by the assignment.

The complexity level of each assignment will vary – each assignment may take several hours to complete. You are expected to start working on them as soon as they are posted so that you have "enough" time to work through the glitches, get the necessary help & still manage to submit on time.

Do not expect me or the TAG to rescue you at the 11th hour! **Late submissions are not encouraged.** If you cannot complete an assignment due to medical condition, send the Doctor note to the professor. You will be given a few additional days to complete the assignment.

Each assignment may vary from small programs to large programs or somewhere in-between. You are expected to spend several hours at a computer every week. The right way to approach the programming assignments is to start on them right away & get help when you get stuck (you can approach the instructor, TAG, or tutors at CS mentor center for help). Do not waste lots of hours trying to fix one specific issue.

Your approach will determine whether programming assignments provide an enjoyable learning experience or end up like a painful activity that ruins your self-confidence.

Your program should be as generic as possible – it should be able to handle all possible valid input values and output meaningful results. You should NOT use any concepts that are not yet covered in the course yet. You should NOT write code just to pass the specific testcases either.

Sophisticated tools are available to detect plagiarism. Suspicious cases will be referred to UTD administration directly - Review http://utdallas.edu/conduct/integrity & http://utdallas.edu/conduct/manage-dishonesty for details.

What you need to do to be successful in this course

Ask for help at any time.
If you do not understand something or are having trouble implementing a concept. The sooner you ask that question; the sooner you will get an answer. That answer will allow you to move forward. I want you to succeed; don't be afraid to ask questions.

- The instructor is available to help during office hours.

Be proactive.

- Don't wait till the day an assignment is due to seek help.
- Please note that I do not have office hours every day.
- If you wait until close to the assignment due date time to seek my help, it is possible that I won't have any office hours that day or there may be many students who will be competing for my time when you use my office hours.
- **You may not email your code to the instructor or grader expecting us to find your errors.**
- Don't wait till the end of the semester to seek help. If you have gotten far behind in your coursework or have done significant damage to your course average, I may not be able to help.

COMPUTER SCIENCE MENTORING CENTER (CSMC)
You can also ask help at CSMC https://csmc.utdallas.edu/

The Computer Science Mentoring Center (CSMC) https://csmc.utdallas.edu/ is a free resource available to all students taking this class.

The CSMC provides assistance in many areas including:

- Understanding core concepts related to this class
- Developing a logical framework for a program
- Connecting programming constructs to the logic of the program
- Assisting in solving syntax and logical errors in your code
- The mentors will meet with you 1-on-1 to address your specific problem areas.
- Their goal is to help you understand what is wrong and how to fix it, but they will not do the work for you.
- For more CSMC information, including location and hours of operation, please view https://csmc.utdallas.edu/

Take responsibility for your education.

- Read your assigned reading before the lecture before course meetings. You are expected to have an understanding of the assigned textbook material before meetings
- Attend every meeting and pay close attention.
- Dedicate 9-10 hours per week outside of class meetings to CS 1337 for reading the book, executing demos, reading the slides, practicing writing code, assignments, and studying for exams.
- I will give challenging assignments to push you toward learning general concepts, developing critical thinking and core programming skills. Part of being a professional is learning how to teach yourself. I am going to guide you through the topics of the semester, but a significant portion of what you take with you to the next class will be things that you have learned on your own.
- The more programs you practice with outside of lecture the better you will do in this course. I will also show you sample demos of programs and of the use of programming constructs / patterns. I will introduce you to program development methodologies. However, you learn to program by doing – coding, testing, and fixing (debugging).
- Enter the sample programs from the text. Experiment by making small changes. Note how the changes affect the program translation and/or execution.
  The more programs you practice with outside of lecture the better you will do in this course. I can teach you the syntax of the C++ programming language and about typical programming constructs. I will also show you samples of programs and of the use of programming constructs / patterns. I will introduce you to program development methodologies. However, you learn to program by doing – coding, testing, and fixing (debugging).
- Complete the Checkpoint questions at the end of the sections of the book text.
- Complete the Review Questions and Exercises at the end of the chapters of the text.

- Pick a few of the Programming Exercises at the end of the chapter and write programs that satisfy the requirements given. This is good practice for the types of coding/MC questions that are on the tests.
  Start your assignment immediately.
  All assignments are designed to be worked on over a period of days.
  You should work on the assignment a little at a time rather than waiting until a day or two before it is due.
  Those that procrastinate will find this class to be much harder than it should be and will face the risk of below average grades.

Practice time management skills.
Good time management is necessary for this class.
- Start your assignment immediately.
- All assignments are designed to be worked on over a period of days. I expect that you will work on the assignment a little at a time rather than waiting until a day or two before it is due. Those that procrastinate will find this class to be much harder than it should be and will face the risk of below average grades.
- Everyone should devote at least an hour a day to this class. Doing this will help you to divide tasks up into chunks and work a little at a time on an assignment rather than waiting until a day or two before it is due.
- You will have a very difficult time succeeding in this class if you schedule to finish every assignment at the last minute.

Attend every class.
You are paying for an education.
Don't waste your money by skipping class.
I will teach you what you need to complete projects and do well on the tests.
You have to be there to get the information.

Don't be afraid to make mistakes!
This is how you learn.
Don't be discouraged when something goes wrong.
Programming takes lots of practice and mistakes will always happen.
Understand the mistakes you made so that you can learn from them for the future.

What each student should expect in this course

A problem solving class.
This class is not a programming class.
Computer science is all about problem solving.
The content of this class is to teach you how to solve problems using a computer.
In order to solve those problems, you will need to learn a new language (C++) and write solutions that the computer can interpret.

An open environment dedicated to learning.
Students should feel free to voice their opinions.
Oftentimes during in class work sessions, I will ask students what they would do in a certain situation.
Each student should feel they can speak freely and also be open to other students to discuss that idea, even if that means that some students will disagree.

Exams focused on application.
Many of the questions on the exam require you to apply your knowledge to answer the question. This may involve finding errors in code or determining output of a code segment. I expect you to apply the knowledge you have learned to the situations on the test. Questions on the test are designed to make sure that you understand what you are doing rather than repeating an example from your notes or the textbook.

A simulated professional experience.
The projects in this class require you to exercise strategies found in "the real world".
Your problem-solving logic for a project may force you to learn new techniques.
You may have to perform code maintenance and improve the efficiency of previously written code.
These challenges offer a small related sample of a typical professional experience.
Once you graduate you may be hired for huge sums of money by a organizations seeking your use these skills.

A solid understanding of C++ and Object-Oriented Programming.
My goal is for you to know all the topics of CS 1337. You should have peace of mind moving on in your program because you will be fully prepared to tackle what the next course in the CS track sequence.

Academic Integrity

All assignments and tests are to be individual efforts.
You are not to collaborate with other students.
Prior to the assignment due date, you are not to:

Discuss assignment solutions with other students, distribute your code to others, or publish your code, discuss assignments on social nets (ex: discord).

Copying of programming assignments or exams, in whole or in part, from other students is considered an act of scholastic dishonesty. Copying assignments from previous semesters is considered an act of scholastic dishonesty.

For some programming assignments, you may use source code provided by the instructor.

You are not to view, copy, or distribute code from any other sources, including code from other students, code from assignments submitted in past semesters, or code from the Internet.

Plagiarism detection software will be employed to detect copying of code.

Grading Concerns:
If you think there is a mistake in the grading of your assignment and would like to request that it be regraded, **you must notify both the grader and the instructor (email the grader and copy the instructor)** of this by email within **one week** after the date the grade is posted in the grade book on eLearning. Your request for any regrade must describe in detail what you perceive as the problem with the grading. A regrade may result in an increase or in a reduction of the original grade.

Most deductions are made because students did not fully read the assignment instructions, did not adequately test their programs, or did not follow the style guidelines provided. You may not change the problem to suit your purposes. Most assignments restrict the use of programming constructs and library functions not covered in lecture, others require that you use constructs or functions. To get the maximum credit you must read the directions carefully and test your programs thoroughly.

Comet Creed

This creed was voted on by the UT Dallas student body in 2014.
It is a standard by which all UTD Student Comets choose to live by and encourage all others to do the same:

"As a Comet, I pledge honesty, integrity, and service in all that I do."

Academic Support Resources

Please go to Academic Support Resources webpage for these policies.

UT Dallas Syllabus Policies and Procedures

The information contained in the following link constitutes the University's policies and procedures segment of the course syllabus. Please go to UT Dallas Syllabus Policies webpage for these policies.

The description, schedules and timelines in this syllabus are subject to change at any time by the discretion of the Professor. Any changes will be announced in classroom session.

Academic Schedule Calendar

All Assignments are due by 11:59 PM on the Assignment Due Date

Week No	Date	Topic/Lecture
01	Jan 21 Mon	Holiday MLK
	Jan 23 Wed	Course, Syllabus, eLearning Review
02	Jan 27 Mon	05 Files, 07 Arrays
	Jan 29 Wed	Continued
03	Feb 03 Mon	08 Searching and Sorting Arrays
	Feb 05 Wed	Continued; Census Due
04	Feb 10 Mon	09 Pointers, Pointers and Arrays, Dynamic Memory
		Assignment 01 Due
	Feb 12 Wed	Continued
05	Feb 17 Mon	10 Chars, C-Strings, String Class
		Assignment 02 Due
	Feb 19 Wed	Continued
06	Feb 24 Mon	11 Structured Data
	Feb 26 Wed	Continued
07	Mar 03 Mon	Continued
		Assignment 03 Due
	Mar 05 Wen	Test 01 Discussion
08	Mar 10 Mon	Continued
		Assignment 04 Due
	Mar 12 Wed	13 Introduction to Classes
	Mar 13 Thr	Test 01 On Campus 10:00 AM 90 Mins TI Auditorium ECSS 2.102 https://map.utdallas.edu/
	Mar 15 Sat	Mid-Term Grades Due on Course Book
09	Mar 17 Mon – Mar 23 Sun Spring Break	
10	Mar 24 Mon	13 Classes, Constructors
	Mar 26 Wen	Continued
11	Mar 31 Mon	14 More about Classes: Copy Constructors, Operator Overload; Appendix D UML
	Apr 02 Wed	Continued
12	Apr 07 Mon	15 Inheritance, Polymorphism, Virtual Functions
	Apr 09 Wed	Continued
13	Apr 14 Mon	Continued
		Assignment 05 Due
	Apr 16 Wed	Continued
14	Apr 21 Mon	16 Exceptions and Templates
	Apr 23 Wed	20 Recursion
15	Apr 28 Mon	Continued, All Make-Ups Are Available
	Apr 30 Wed	Test 02 Discussion
	Apr 26 Fri	Assignment 06 Due (no late days)
		All Make-Ups Due (no late days)
16	May 07 Wed	Test 01 On Campus 10:00 AM 90 Mins TI Auditorium ECSS 2.102 https://map.utdallas.edu/
	May 17 Sat	Final Grade Viewable Online on Blackboard, Check your grade.
	May 21 Wen	Final Grades Due Posting to Orion

The professor reserves the right to change any part of this syllabus, including this schedule calendar.
Any changes will be announced in class and posted to eLearning site.