

COURSE INFORMATION:

Course Number: CS 1337.002/005
Course Title: Computer Science I
Course Term: Spring 2026
Schedule: M W 10 - 11.15AM, 11.30AM - 12.45PM
Location: ECSS 2.201/2.305

INSTRUCTOR CONTACT INFORMATION:

Name: Srimathi Srinivasan
Email Address: Srimathi.Srinivasan@utdallas.edu
Office Location: ECSS 3.608
Office Hours: Mondays, Wednesdays 1 - 2 PM in person/Teams
[Click here to enter my Office room](#)

GRADER CONTACT INFORMATION:

TA:
TA email:
Office hours:
Location:

Email: When you send a mail to me or TA, please **specify 1337.002/1337.005 in the subject of the email.**

COURSE PREREQUISITES:

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

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. The programming language of choice is C/C++. Students will also be registered for an exam section.

CLASS PARTICIPATION:

*Regular class participation is expected regardless of course modality. Students who fail to participate in class regularly are inviting scholastic difficulty. **Regular attendance is highly recommended.*** Three consecutive absences or more than five absences lead to a letter grade drop. Seven absences lead to an F. However, **two attendances will be excused/dropped** when computing the final grade to cover the common issues like car-trouble, oversleep, mild sickness. However, if you provide a doctor's medical certificate your absence will be excused. No other reasons will be accepted for an absence excuse.

CLASS MATERIALS:

The instructor may provide class materials including the syllabus, slides, and sample programs will be shared though **UTD box folders** registered for this class as they 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 Office of Student Accessibility accommodation. Failure to comply with these University requirements is a violation of the [Student Code of Conduct](#).

STUDENT LEARNING OBJECTIVES/OUTCOMES:

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

1. Ability to implement simple searching and sorting algorithms.
2. Ability to implement pointers and perform simple memory management.
3. Ability to implement structured data types.
4. Ability to define and implement a class.
5. Ability to use fundamentals of object-oriented design.

Notes: 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)

Course Learning Outcome	Homework	Exams
1. Ability to use single and multi-dimension arrays	IC1, HW1	1
2. Ability to implement simple searching and sorting algorithms.	HW2	1
3. Ability to implement pointers and perform simple memory management	HW3,4,5	1
4. Ability to implement structured data types.	HW6,7	1
5. Ability to define and implement a class.	HW8,9,10	2
6. Ability to use fundamentals of object-oriented design.	HW11,12, Project	2

REQUIRED TEXTBOOKS AND MATERIALS:**Required Textbook:**

Starting out with C++. From control structures through objects, Ninth Edition, by Tony Gaddis, Pearson Education, Inc. ISBN: 978-0-13-449837-9.

ZyBooks:

Zylabs is mandatory and it will cost \$45. Labs, in class activities, assignments and practice problems will be posted in Zylabs and you will post the solutions there. It is an auto grader tool. The feedback on your code will however be provided to you by the graders.

1. Click any zyBooks assignment link in your learning management system (Do not go to the zyBooks website and create a new account)
2. Subscribe

A subscription is **\$45**. Students may begin subscribing on Jan 06, 2026 and the cutoff to subscribe is Apr 25, 2026. Subscriptions will last until May 22, 2026.

Software Options:

- Windows: Code::Blocks IDE from codeblocks.org/downloads/binaries - codeblocks-20.03mingw-setup.exe (mingw is important – it is the compiler!) - very easy to install!
- If your compiler is not compiling go to settings → compiler → Toolchain executables → autodetect
- Mac: use pre-installed XCode
- Web based: Online GDB compiler onlinegdb.com/online_c++_compiler (includes debugging!)
- There are several other popular ones out there, including Visual Studio - incredibuild.com/blog/best-c-ides has a good list!

Additional optional resources:

C++ language tutorial www.cplusplus.com/files/tutorial.pdf
 C++ reference: www.cppreference.com
 C++ tutorial www.learncpp.com

I will use code blocks. codeblocks-17.12mingw-setup.exe can be downloaded from <http://www.codeblocks.org/downloads/26>

Important Dates and Times	<ul style="list-style-type: none"> • First day of class: Wednesday, January 21st, 2026 • Exam 1: Wednesday, February 25th, 2026 10% weightage • Exam 2: Friday, March 13th, 2026 25% weightage • Exam 3: Friday, May 8th, 2026 25% weightage
----------------------------------	---

Exam1 will be conducted in the classroom. Exam 2 and 3 will be in allocated rooms. This will be announced one week prior to the exam.

Tentative Course Calendar

Week	Dates (M W)	Class Activity/Notes	ICs Assigned	HWs Assigned	Practice Due	Chapters
1	Jan 21 st	Review of syllabus		HW0		
2	Jan 26 th , 28 th	Review of 1436 Files/arrays	IC1, IC2	HW1		Ch 1-7
3	Feb 2 nd , 4 th	Search and Sorting	IC3, 4	HW2_S		Ch 8
4	Feb 9 th , 11 th	Pointers	IC5	HW3_P	P1 02/16	Ch 9
5	Feb 16 th , 18 th	Pointers, C Strings	IC5b, IC6, 7	HW4_P		Ch 10
6	Feb 23 rd , 25th	C Strings, Exam 1	IC8, IC9	HW5_Strings		Exam 1: We classroom 10%
7	March 2 nd , 4 th	Structured Data	IC10, 11	HW6_stru	P2 03/09	Ch 11
8	March 9 th , 11 th March 13th	Introduction to classes, Exam Review		HW7_stru		Exam 2: Friday 6 PM 25% weightage
9	March 16th, 18th	Spring break				
10	March 23 rd , 25 th	Classes, Constructors, Mid Semester review	IC 12	HW8_C		Ch 13

11	March 30 th , April 1 st	Overloaded constructors, UML, Copy constructor	IC 13	HW9_BA	project assigned	Ch 14
12	April 6 th , 8 th	Operator Overload, Complex	IC 14,15	HW10_OO		Ch 15
13	April 13 th , 15 th	Inheritance, Polymorphism,	IC 16	HW11_Agr		Ch 15
14	April 20 th , 22 nd	Pure Virtual Functions, Project		HW12_In	P3 04/27	Ch 16, 20
15	April 27 th , 29 th	Exceptions, Recursion	IC 17		Project 05/03	project due Ma
16	May 4 th , 6 th May 8th	Linked Lists, Exam Review				Exam 3: Friday 4-6 PM 25% we

Course Prerequisite for CS 2336/2337: CE/CS 1337 with a grade of C or better or equivalent. You need to get a minimum of 70%(c) by taking an average of both the exams to proceed to the next course.

Letter grades will be assigned as follows:

97-100	A+	94-97	A	90-94	A-
87-90	B+	84-87	B	80-84	B-
77-80	C+	74-77	C	70-74	C-
67-70	D+	64-67	D	60-64	D-
Below 60	F				

GRADING POLICY:

Grading Criteria	<p>In class participation: 10% Exam 1: 10%, Exam 2: 25%, Exam 3: 25%, Programming Assignments: 20, A small project 10%.</p> <p>(To enroll in course CS/CE 2336, you must have completed either CE/CS 1337 with a grade of C or higher). So, you need to score an average of 70% (C-) across both exams 2 and 3 to qualify for a C or better in the course. For eg. if your final score is say B- but your exams score is D, you final score will still be a C-. You can pass the course but may not be eligible to move to 2336.</p> <p>Historically students who skip programming assignments, or do not put much effort into their programming assignments, or get a lot of help from classmates, mentors, or others, do not perform well on exam questions testing the material covered by the assignment.</p>
Programming Assignments	<p>Programming assignments are given every week. Weekly Assignments will be assigned on Wednesdays and are due by next Tuesday at midnight. They will have equal weightage – all of them together will contribute 20% to the final weighted grade. This high weightage indicates the importance of assignments in this course!</p> <p>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 TA 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.</p>

Each week's assignment may vary from multiple small programs to one large program 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, TA, or tutors at CS mentor center for help). Do not waste lots of hours trying to fix one specific issue. In simple words, your approach will determine whether programming assignments provide an enjoyable learning experience or end up like a painful activity that ruins your self-confidence.

You can develop & test your program using any C++ IDE, but you need to pass all the official testcases in Zylabs to complete. Assignments are auto graded through Zylabs & the scores will appear automatically in elearning. Your program should be as generic as possible – it should be able to handle all possible valid input values and output meaningful results. As time permits, TA will manually review your code in Zylabs and give feedback. All submissions are subject to random manual inspection as well - you should NOT use any concepts that are not yet covered in the course yet. You should NOT write code just to pass the specified testcases either. Your assignment score may be reduced to 0 for such violations.

We are all here to learn! Sophisticated tools are available in Zylabs and beyond 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.

Diagnostic Tests

1. **Mandatory Diagnostic Tests:** All students enrolled in the 1337 course must take the diagnostic tests.
2. **Purpose of Test Results:** The results will help students understand their competency levels. This approach will ensure all students reach the same level of competency. Based on their scores, they will be required to attend sessions in CSMC.
3. This test will be part of your first assignment. You will need to attend the **remedial Sessions in CSMC based on your test score:**
 - **Above 90%:** Students don't need to attend any session, and they automatically receive 100% in their assignment.
 - **80-90%:** Students must attend one session in CSMC to receive 100%.
 - **70-80%:** Students must attend two sessions in CSMC to receive 100%.
 - **60-70%:** Students must attend three sessions in CSMC to receive 100%.
 - **Below 50%:** Students must attend all sessions in CSMC to receive 100%.
4. CSMC will give us an update on the sessions attended and your assignment score will be updated accordingly
5. **Remedial Sessions Timing:** CSMC will conduct the remedial sessions during the 2nd and 3rd weeks of the semester. All students are required to attend these sessions based on the above criteria.

In class Exercises

There will be 1 or 2 activities every week (relatively simple compared to Assignments - should not take more than 1 hour each) to ensure that you are keeping up with the class content. You also need to complete an in-class activity in every class - bring your laptop to every class - all of them will contribute equally & together they will account for 20% of your final grade.

Activities ensure that you are keeping up with the course - they cannot be made up after the deadline! In other words, late submissions will NOT be accepted, however **one lowest score** will be dropped when computing the final grade, to cover the common issues like car-trouble, oversleeping, etc. If you cannot complete an activity due to medical condition, send the Doctor note to the professor. You will be given additional time to complete.

	Exercises may be given in lectures with or without previous notification. There is no make-up for these. It must be submitted by next day 11.59PM of the lecture day.
Make-up Exams	Make-up examinations will be administered only for well-documented emergencies . A student must make every attempt possible, via email, to notify the instructor that he/she will miss a scheduled exam prior to the scheduled date and time or immediately thereafter. If notification is not received in a timely manner, no make-up will be given.
Project	There will be one final project. You will implement the project using Object Oriented Design concepts. More details will be provided in the middle of the semester.
Late Work	Late submissions will not be encouraged for Programming assignments or in class activities as the solutions will be discussed in class the next day. However, I found out that many students put in an effort and are not able to finish the assignment within the deadline or if a student is stressed out, a few days late submission will be allowed, but the grading will differ. You don't have to send a mail on this regard, asking for permissions for late submissions. Your assignments will be graded as below if submit by Tuesday: 100% Wednesday: 90% Thursday: 80% Friday: 70% After that: 0%
Class Attendance	Regular attendance is highly recommended. Three consecutive absences or six absences lead to a letter grade drop. Eight absences lead to an F. However, two attendances will be dropped when computing the final grade to cover the common issues like car trouble, oversleep, mild sickness and religious holidays. You don't have to send a mail on this regard, asking for permission to take off. If you are sick, your absence will be excused only if you give a doctor's note. This will not be counted against your 6 absences for a letter grade drop. Your 6 or 8 absences for the letter drop include the two dropped attendances. If a student attends all classes he/she will get two bonus points over the final grade.
Classroom Citizenship	The instructor encourages students to take active part in class discussions. No question is too simple/stupid to be asked. So, do not hesitate. Use of smart phones and headphones/earphones is strictly prohibited. Laptops need to be closed, until I request to open for an in-class activity.
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 http://go.utdallas.edu/syllabus-policies for these policies.

Exam Planning and Policies

Exam Schedule

- **Exam 1:** February 25 (Wednesday) during regular class hours.
- **Exam 2:** March 13, 4:00–6:00 PM (rooms scheduled).
- **Exam 3:** May 8, 4:00–6:00 PM (rooms scheduled).

Exam Content

- **Exam 1 Topics:**

- Files and Arrays
- Searching and Sorting
- Pointers
- **Exam 2 Topics:**
 - Searching and Sorting
 - Pointers
 - Strings
 - C-Strings
 - Structured Data

Weightage and Score Adjustment Policy

- **Exam 1** will serve as a *wake-up call* for students.
- If a student performs poorly on Exam 1 but performs very well on Exam 2, the weightage will be adjusted to:
 - **Exam 2:** 35% (instead of 25%)
- If a student performs well on Exam 1 but only average on Exam 2, the weightage will be:
 - **Exam 1:** 10%
 - **Exam 2:** 25%

Passing Criteria

- Students must earn at least a **C** to progress to the next level (courses 2336 or 2337).
- Students must achieve a **combined 70% average across Exam 2 and Exam 3**.
- Since Exams 2 and 3 together total 50 marks, students must score a **minimum of 35 marks** to move to 2336/7.
- If Exam 2 carries **35% weightage**, Exams 2 and 3 together total 60 marks, students must score a **minimum of 42 marks** to move to 2336/7.
- The final calculation will use **70% of the average of Exam 2 and Exam 3** to decide if he can move on to 2336/7.
- **New Policy:** Students must earn **at least a C- average on Exams 2 and 3** to be eligible to progress, regardless of their assignment performance. While a **C** remains the required overall course grade, students who earn a higher overall grade (for example, a B) but **do not achieve at least a C- on the exams** will have their **final course grade adjusted to C-**.

Assignments will be graded on a 100-point basis, utilizing the following criteria:

		Max Score
Pseudocode	High level explanation of steps at the top of program/each function	10%
Source Code	Overall design of the program	40%
Source Code	Assigning meaningful names for variables and functions	10%
	Formatting and coding style	10%
Execution	Test cases	30%
Total		100%

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 till close to the assignment due date 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 come to my office hours. **You may not email your code to the instructor or grader expecting us to find your errors.**
 - You can also ask help at CSMC <https://csmc.utdallas.edu/>
 - 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)

The Computer Science Mentoring Center (CSMC) 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
- Exam reviews and reworks (by faculty request)

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 information about

the CSMC, including location and hours of operation, please visit <http://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, watching recordings, practicing writing code, assignments, exercises 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 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).
- 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 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 I will ask you on the exam.
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.

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.
- I expect everyone to 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 give you everything you need to complete projects and do well on the tests. You have to be there to get the information.

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. Study 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 foreign language (C++) and write solutions that the computer can interpret.

An open environment dedicated to learning. I want students to feel free to voice their opinions. Oftentimes as we code in class, I will ask students what they would do in a certain situation. I want each student to feel as if he/she can speak freely and also be open for other students to discuss that idea, even if that means that some students will disagree.

Exams focused on application. Many of the questions I ask on an 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 logic for a project may force you to learn new techniques that haven't yet been discussed in class. You will have to perform code maintenance and improve the efficiency of previously written code. These things offer a small taste of how life might be once you graduate and are given large sums of money by a company seeking your skills.

A deep 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 sequence will throw at you.

Academic Integrity:

All assignments, exercises and exams 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. Copying of programming assignments, exercises, or exams, in whole or in part, from other students will be considered an act of scholastic dishonesty. Copying assignments from previous semesters will be considered an act of scholastic dishonesty.

For 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 or exercise 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. Keep in mind that 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 that Comets choose to live by and encourage others to do the same:

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

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 <http://go.utdallas.edu/syllabus-policies> for these policies.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.