Course Syllabus for CS/CE 1337.003 - Computer Science I

COURSE INFORMATION:

Course Title	CS/CE 1337 - Computer Science I
Section	003
Course Term	Fall 2023
Class meeting times	Tuesdays & Thursdays between 10:00am-11:15am
Classroom	ECSS 2.306
Course modality	This course is scheduled to be **taught in the Traditional (In person) Mode**. Should it become necessary to meet online, meetings will be held in Teams in the appropriate Lecture Channel. Online meetings will be recorded. Check the course Announcements and your UTD email account for updates.

INSTRUCTOR CONTACT INFORMATION:

Name	**Priya Narayanasami**
Contact	**Only via Teams one-on-one chat stating the course number and section number.** It is likely that messages sent on the weekend (after 5pm on Friday) may not be responded to until Monday
Office Hours	**Online office hours link :** **T : 5pm - 6pm & W : 10am – 11am** (Reach out by Teams to set up an appointment if these hours do not work out for you) **Do not procrastinate. Historically, my office hours become significantly busier as we near a due date.** *Poor planning on your part does not necessitate an emergency on mine. – Bob Carter*

TA CONTACT INFORMATION:

Name	**TBA**
Contact	**Only via Teams one-on-one chat stating the course number and section number.**
Office Hours	**Online office hours link : TBA**

CS Mentor Center Contact Information :

Approach the CS Mentor Center for help.

They are available during most of the days; be aware they tend to be crowded during exam times – they operate on a first-come first-served basis.

COURSE DESCRIPTION, PREREQUISITES AND COREQUISITES:

CATALOG DESCRIPTION:

(3 semester credit hours) 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.

Prerequisite: CS 1336 with a grade of C or better or equivalent. (Same as CE 1337) (3-0) S

COURSE LEARNING OBJECTIVES/OUTCOMES:

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

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

Notes: This course is the middle one in the UTD CS programming sequence. The 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 1336 Programming Fundamentals (C++)
CS 1337 Computer Science I (C++)
CS 2336 Computer Science II (Java)

REQUIRED TEXTBOOKS AND OTHER MATERIALS:

Required Textbook:

- Starting Out with C++, From Control Structures through Objects (9th edition); Gaddis, Tony; Addison-Wesley Publishing. ISBN-13: 978-0134498379 ISBN-10: 0134498372
- **Any version - Seventh or newer - is also acceptable.**
- As you read the text, watch the corresponding Video Notes; available at http://www.pearsonhighered.com/gaddis/.
- See the **TENTATIVE CONTENT COVERAGE SCHEDULE** later in the document for suggested reading schedule.

Required Subscription to Codio :

- We will be using Codio platform to submit & auto-grade the assignments and most activities.
- Your submissions will be tested against several testcases, similar to ZyLab platform you might have used in CS 1136/1336 courses.
- You will be promoted to purchase a license ($40 fee) as part of the first assignment/activity we will do on that platform.

Computer with Compiler:

- This is a programming course and I believe in hands-on programming**.**
- **Please bring a laptop for completing in-class activities.**
    - **For information about the Student Technology Initiative including information on financial aid and technology loaner program see:** https://oit.utdallas.edu/technologyinitiative/
- 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!
    - Windows: VS Code
    - Mac: use pre-installed XCode
    - Web based: https://replit.com/new/cpp - this popular site supports several languages!
    - 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!
    - Irrespective of the compiler you use for development, your grade will be based on passing the auto-graded tests run in Codio.
    - You will not get credit for tests failed in Codio, regardless of the results on your machine.

Additional optional resources:

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

CLASS MATERIALS:

- Instructor's PPTs and code done in class : Click on Prof Box folder link in the eLearning system.
- Other materials including the syllabus, quizzes & tests, assignments, and in-class activities/participation exercises, etc. will be posted and or linked in eLearning. https://elearning.utdallas.edu

ACADEMIC CALENDAR (TENTATIVE, but the test dates are fixed):

Week	Topic	Suggested reading schedule
1 - 4	Syllabus review & Codio Review: concepts covered in 1336 – decisions, loops, arrays & functions, debugging Problems from coding contest sites (USACO and Kattis)	Ch. 1-8
5	Pointers & Review C language	Ch. 9
6	Characters, C-Strings, and the String class	Ch. 10
10/05/2023	Test 1 in UTD testing center. Please **register for the test and submit the screen shots in Book_Slot for test 1 ; found under Activities and the due date for those activities is 08/22/2023 before 11am**	
8	Structured Data (struct)	Ch. 11.1-11.10
9 - 13	Introduction to Classes & Object-Oriented Design	Ch. 13 - 15
11/21 & 11/23	Thanksgiving Holiday	
14 & 15	Introduction to Recursion	Ch. 19
12/07/2023	Test 2 in UTD testing center. Please **register for the test and submit the screen shots in Book_Slot for test 2 ; found under Activities and the due date for those activities is 08/22/2023 before 11am**	
12/20/2023	Final grades	

GRADING POLICY

Extra credit work : None.

Summary :

- Exam/Test : 40%
- Homework / Assignments : 30%
- In-class activities / Attendance : 30%
- Don't ask for a bump in the final grade - it should be earned, not just given!

Tests : 2 Tests (equal weights)

- Look into the Academic calendar given above for date, time for slot registration & other related instructions.
- Exams will be administered in the testing center.
- Ensure that both of the Book_Slot activities are done in the first class itself.
- **A make-up examination will not be scheduled/administered for students without an exam reservation with the testing center, regardless of the reason/excuse supplied for missing the examination.**
- **Closed book and closed notes – cheat sheets are not allowed.**
- Each test will be comprised of multiple-choice questions, fill-in-the-blank questions , true/false questions, find the error(s) in the given code snippet, find output of the given code snippet, and around 3-5 coding questions.
- You will be required to code in the tests.
- To avoid the need for memorization, necessary documentation or sample code will be provided.
- **Make-up examinations** :
  - will be administered **only for well-documented emergencies** (Doctor's note / hospital receipt / a coach's note will be required as proof.)
  - A student must make every attempt possible, via Teams one-on-one chat, to notify the instructor that they will miss a scheduled exam **prior** to the scheduled date and time or **immediately** thereafter. **If notification is not received in a timely manner, make-up will not be given.**
  - Makeup test due to other scenarios will result in 20% penalty. Also, make-up test questions will be different, so the complexity may vary a bit.
- The **best way to prepare for the test** will be to
  - attend each class & **be engaged.**
  - complete the in-class activities.
  - complete the weekly assignments.
  - get your doubts clarified in a timely manner.
  - work out problems dealing with our concepts from Kattis to get more practice.
- **Professor** will be grading the tests.
- **Questions about the test grade?**
  - **notify the Professor using Teams one-on-one chat within 3 days** after the date the grade is posted in the grade book on eLearning.

Homework / assignments:

- **Equal weight for all the assignments - 40%**
- Every week an assignment is due on Wednesday @ 11:59pm
    - Week #1's homework will be due on 08/30/2023 @ 11:59pm.
- The complexity level of each assignment will vary:
    - multiple small programs to one large program OR somewhere in-between
    - 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.
- The right way to approach the programming assignments is :
    - to start on them right away. **Do not procrastinate.**
    - get help when you get stuck (you can approach the instructor, TA, or tutors at CS mentor center for help).
    - Do not waste many 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 may not send your source code to the TA or instructor unsolicited by Teams expecting us to debug/fix it.**
    - This is not reasonable. There are too many of you for us to do this. Also, part of learning to program is developing your own debugging skills. It is your responsibility to develop your code in a manner that minimizes errors.
    - **The professor/TA will help you find errors during office hours**, but you should have narrowed down the problem before asking for help.
    - *When* **they** *look at your code, they will expect to see debug statements, to see that you have done your best to locate the error(s). If they do not see these, they will ask you to check back after you have added them.*
    - Do not expect the instructor or the TA to rescue you at the 11th hour!
        - **Historically, the office hours have become significantly busier as we near a due date.**
        - *Poor planning on your part does not necessitate an emergency on mine. – Bob Carter*

- **<u>Late submission is not accepted.</u>**
    - Link to submit the assignment on eLearning will disappear after the due date/time.
    - Outside of eLearning submission is not accepted – do not email or submit via Teams because you cannot find the link to submit.
    - If you cannot complete an assignment due to a medical condition, send the Doctor note to the professor using MS Teams one-on-one chat. We expect at least a small portion to be completed as the expectation is that you started early and in the middle you fell sick. Depending on how much you have progressed, you might be given a few additional days – the discretion is left to the professor. If you wait until the 11th hour, you will not have anything to submit and there is nothing much that can be done at that time.
- Homework needs to be accessed via eLearning only; that link will take the students into Codio where the student can read about the problem in detail; the student may choose to code in Codio or in any IDE; if done outside of Codio, the student needs to paste the solution into Codio and run the code at least once before the deadline. Codio maintains versions of the submission. That will help in determining the progress made by the students.
- Assignments are auto graded through Codio & the scores will appear automatically in eLearning.
    - Your program should be as generic as possible – it should handle all possible valid input values and output meaningful results.
    - As time permits, TA will manually review your code in Codio 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.
    - You should NOT write code just to pass the specified testcases either. Your assignment score may be reduced to a 0 for such violations.
    - **TA will be responsible for grading all types of assignments and in-class activities.**
    - **Questions about the assignment grade?**
        - **notify the <u>Professor & the TA</u> using Teams group message within 3 days** after the date the grade is posted in the grade book on eLearning.
- We are all here to learn! Sophisticated tools are available in Codio 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.

Activities : 20 %

- There will be 1 or 2 activities every week (simple compared to Assignments - should not take more than 1 hour each) to ensure that you are keeping up with the class content (complete tutorials at home, finish simple exercise or take online quiz, etc.)
- You also need to **complete an in-class activity in every class - bring your laptop to every class.**
- All of these will contribute equally & together they will account for 30% 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, the **two lowest scores** will be dropped when computing the final grade, to cover the common issues like car trouble, oversleep, etc. You should be able to earn the full 30% if you attend all the sessions and submit the dues for that class on time.
- If you cannot complete an activity due to a medical condition, send the Doctor note to the professor using MS Teams chat. You may be given additional time to complete it OR exempted from that activity.

Letter grades will be assigned as follows:

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

You have to earn the grade with your work. There are no plans to round up to improve your final grade.

Weighted total in your gradebook shows the current weighted grade based on your graded work. For example, if you have done only 2 assignments & 2 weeks of Class work so far, the current grade will be based on only those entries. So, it will continue to change throughout the semester as the items are graded.

Course & Instructor Policies

- **Teaching Philosophy Statement**

    - My goal is to make each class as enjoyable as possible! I use a project-based learning approach to teach. In other words, I work on a series of real-world scenario-based problems. After discussing each problem at a high-level, I expect the students to code along with me in the class.
    - Since I focus on the big picture and critical thinking skills, I encourage the students to read the book to get the complete details.
    - Students are expected to bring a laptop to each class; they are NOT allowed to do anything in the laptop/phone unrelated to the class work. Student(s) might be asked to leave the class if this behavior is observed on a consistent basis and their attendance for that class might be voided.
    - MS Teams one-on-one chat is the preferred means of communication; please state the course # & section #; without that info, reply will be severely delayed.
    - It is possible that a message received after 5pm on Friday may be responded on Monday after 9am.

- **Academic Integrity:**
    - **All assignments, exercises, and exams are to be individual efforts.**
    - You are not to collaborate with other students.
    - Prior to the assignment/quiz/exercise/exam due date , you are expected not to discuss solutions with other students in anything but the most general terms (for quizzes/exercises/exams you may not discuss at all), distribute your code to others, or publish your code.
    - Copying programming assignments/quizzes/exercises, in whole or in part, from other students will be considered an act of scholastic dishonesty.
    - Copying assignments/quizzes/exercises from previous semesters will be considered an act of scholastic dishonesty.
    - You are not to view, copy, or distribute code from any other sources, including code from other students, code from assignments/exercises/quizzes submitted in past semesters, or code from the Internet.
    - Plagiarism detection software will be employed to detect copying of code.
    - **Codio includes built-in comparison software that compares your submission to every other submission in the course.**
    - ***Caution:*** ***Do not share your code*** with one of your peers so they can "learn from it". They may submit it as their own. They may share it with others. ***You*** are guilty of academic dishonesty if you give your code to others or publish it in chat rooms or on websites and another student submits it as their own work.
    - Falsifying output/test results is academic dishonesty. Your program must include the code that does the processing/calculations/work required to generate the results/outputs.

Extra Credit Work: None

Student responsibilities:

- You are responsible for all the material in the assigned reading in the required course textbook.
- You are responsible for all the material on the slides.
- You are responsible for all material discussed in course meetings.
- You are responsible for all material supplied on eLearning (including announcements and due dates of all homework assignments/activities.
- Students are expected to be respectful of each other and of the course instructor.
- Disruptive behavior (Non-academic use of technology for this course, talking while lecture is on, etc.,) will not be tolerated. Student(s) might be asked to leave the room and they might lose attendance for that day.

What you need to do to be successful in this course:

- Attend every course meeting and pay close attention.
- Read your assigned reading before the lecture. The instructor expects you to have completed this introductory material before lecture meetings.
- Review the slides/slide recordings if you have questions about the reading and review them again before the exam.
- Dedicate **10 -12** hours per week **outside of attending class** for reading, watching recordings(when available), practicing writing code, completing assignments and labs, and studying for exams.
- Do all your work and do it yourself. Students who get too much help from others: the instructor, mentors, peers, and the internet do not have the knowledge they need to successfully complete the examinations. Getting help is ok but the student needs to understand and retain the knowledge as to why the program is written that way.
- **Do not procrastinate**. Initial assignments are typically easy, but as we progress through the course the assignments will become more challenging. All assignments are designed to be worked on over a period of *days or weeks*. Start early so that you have time to try alternate approaches, ask questions, and test your program. A program is not complete when it compiles. Your program must produce correct results under various conditions. You must design test cases in addition to designing your code.
- Learn to debug your programs yourself.
    - Add print statements that print the values of inputs and the results of intermediate calculations.
    - Add print statements to indicate that sections of code have been reached during execution.
    - Learn to use the integrated debugger.
- Do not wait till the end of the semester to seek help. If you wait until late in the semester, it is difficult to catch up, as the course is constantly moving forward.

- If you do not understand a concept: Reread the text, review the slides, look into the sample programs that are inside the Professor's Box Folder. Plan to attend the office hours with a list of specific questions. Visit the Computer Science Mentor Center (in-person or online sessions) with specific questions.
- Write more programs than are required. The more programs you practice with outside of lecture, the better you will do in this course and beyond. The professor can teach you the syntax of the C++ programming language and about typical programming constructs. The professor can & will show you samples of programs and of the use of programming constructs / patterns. The professor can introduce you to program development methodologies. However, one learns to program by doing – coding, testing, and fixing (debugging). This course is like a mathematics course – you need to work on **many** problems.
  - Enter the sample programs from the text. Experiment by making minor changes. Note how the changes affect the program translation and/or execution.
  - Select some of the Programming Exercises at the end of the chapter or in Codio and write programs that satisfy the requirements. This is good practice for the types of coding questions that would be asked in the exams.
- Make sure you know the answers to the Checkpoint and Review Questions found in the text.

MORE FROM UTD:

Class Materials

The instructor may provide class materials that will be made available to all students 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 AccessAbility accommodation. Failure to comply with these University requirements is a violation of the Student Code of Conduct.

Classroom Conduct Requirements Related to Public Health Measures

UT Dallas will follow the public health and safety guidelines put forth by the Centers for Disease Control and Prevention (CDC), the Texas Department of State Health Services (DSHS), and local public health agencies that are in effect at that time during the Fall 2023 semester.

Class Attendance

The University's attendance policy requirement is that individual faculty set their course attendance requirements. Regular and punctual class attendance is expected. Students who fail to attend class regularly are inviting scholastic difficulty. In some courses, instructors may have special attendance requirements; these should be made known to students during the first week of classes.

Class Participation

Regular class participation is expected. Students who fail to participate in class regularly are inviting scholastic difficulty. A portion of the grade for this course is directly tied to your participation in this class. It also includes engaging in groups or other activities during class that solicit your feedback on homework assignments, readings, or materials covered in the lectures (and/or labs). Class participation is documented by faculty. Successful participation is defined as consistently adhering to university requirements, as presented in this syllabus. Failure to comply with these University requirements is a violation of the Student Code of Conduct.

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."

Academic Support Resources

The information contained in the following link lists the University's academic support resources for all students.
Please see http://go.utdallas.edu/academic-support-resources.

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 review the catalog sections regarding the credit/no credit or pass/fail grading option and withdrawal from class.

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.