CS 1337 Computer Science I
Spring 2021 Syllabus UTD

INSTRUCTOR INFORMATION

Name	Jason Smith
E-mail Address	smith@utdallas.edu
Office	ECSS 3.232
Office Phone	972-883-4835

Office Hours:

M/W 2:00 – 3:30 PM
Tu/Th 2:30 – 3:30 PM
No appointment necessary; just stop by.
All office hours will be held virtually through Blackboard Collaborate. Please use the link provided in eLearning

Questions: All questions must be in person (class time and office hours) or posted to Piazza (http://piazza.com/utdallas/Spring2021/cs1337)

COURSE INFORMATION

Course Number	CE/CS 1337.007 and 1337.008
Credit Hours	3
Meeting Time	Tu/Th 10:00 – 11:15 AM (1337.008) 1:00 – 2:15 PM (1337.007)
Room	All classes will be held through Microsoft Teams at the regular class time. Students unable to attend may download recordings of the lecture.

ASYNCHRONOUS OPTION

Any student may elect to take this class asynchronously at any time during the semester. **Students choosing the asynchronous option must email me at least 48 hours prior to the first asynchronous class.** This will give me time to adjust the attendance settings. If you elect the asynchronous option, please note that the due dates and times still apply to you. The biggest difference will be how/when you view the lectures. The lectures will be given in Microsoft Teams and the recordings of all lectures will be posted in Microsoft Stream. A link to Microsoft Teams and Microsoft Stream is available in eLearning.

Exams will be given during the scheduled class time. **If you are an asynchronous student and the time of the exam conflicts with your schedule, you may schedule your exam at a different time on the same day. Arrangements for a different time must be made at least 48 hours in advance of the day of the exam.** Once the exam starts, the exam must be completed within the allotted time limit

DO YOU NEED ASSISTANCE?

Problem Solving Procedure

1. Try to solve it yourself – use the internet to research the problem and try different solutions. If you can't solve it after a couple of hours move to the next level.

2. Consult with your classmates – Post your question in Piazza (or the class GroupMe or Discord server). On Piazza, you will be able to post your questions (anonymously if you wish) about

anything related to the class and get a response either from me or a classmate. Private posts (that only I can see) should be reserved for grade related questions or questions involving large snippets of code. Since there are multiple people that can answer questions, you should get a quicker response allowing you to complete the tasks you are working on. If your question can't be answered there, move to the next level.

3. Visit the CSMC – If your classmates can't answer the question, check with a mentor at the CSMC (https://csmc.utdallas.edu). If the CSMC mentors can't answer your question, move to the next level.

4. Visit me during office hours – If you are truly stumped, I will give you the information you need to move forward.

Help Desk: For help with issues regarding your computer, UTD maintains a walk-in help desk. Visit their Web site for details: http://www.utdallas.edu/ir/helpdesk/

If you need help, please make the effort to reach out. We can't help you if we don't know that you need help.

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

WHAT DO I NEED FOR CLASS?

If you need help obtaining any of the required class resources below, please contact me. I will help you get access to the required tools.

- **A Webcam (Required):** This course will use Honorlock – an online exam proctoring tool. To successfully take an exam, you must have a web camera with microphone, a laptop or desktop computer (no tablets/phones), Chrome browser, a reliable internet connection and a photo ID (anything with your picture and name). Before starting the exam, you will be prompted to install the Honorlock Chrome Extension (which you can remove after you finish the test) if it is not already installed. You will then go through the authentication process. The web camera will monitor you throughout your test. Please see the Testing Guidelines and Support Information for additional information.

  You do not need a fancy webcam. Webcams can be purchased for less than 30 dollars on Amazon. If your laptop has a webcam, you do not need to purchase a different one.

- **Zybooks (Required):**
  - CS 1337 – Computer Science I
  - 978-1-394-06952-1
  - All projects will be submitted through Zylabs for grading and plagiarism checking

- **Textbook (Required)**: Exams will be based on the material in this book.
  - **Option 1:** Revel System
    - Revel for Gaddis C++ - Access Code (Digital Book)
      Gaddis, Tony; Pearson Publishing
      ISBN 0-13-449837-9

    - The access code is for a digital version of the book using the REVEL platform. The REVEL platform provides interactive elements to the digital textbook such as

animations, videos and coding samples that readers can modify and execute.

- In past semesters, students have stated that the interactive textbook helped them learn the material better

- Students using this option do not need to buy the physical version of the book

- **Option 2**: Physical book
  - Starting Out with C++, From Control Structures through Objects (9th edition – grapefruit slice) ; Gaddis, Tony; Pearson Publishing ISBN 0-13-449837-9

  - As you read the text, watch the corresponding VideoNotes. The VideoNotes are available at http://www.pearsonhighered.com/gaddis/.
    NOTE: VideoNotes are only available with an access code. If your book does not have an access code, you can buy one online at the above address. **The access code is not required for class**, but some of you may find the material accessible with this code to be a good resource.

- Students may use the 8th edition of the book. In doing so, students accept the responsibility of reading and learning the topics not present in the 8th edition. I will not copy any information from the 8th edition for students, although you are free to take pictures of pages from my book during office hours.

- Many students have found this book online in pdf format for free. If you have access to a free version of the book, please take advantage of it.

- **C++ Compiler (Required)**
  - I encourage everyone to use Online GDB (https://www.onlinegdb.com/online_c++_compiler)
    - This will better simulate the ZyBooks environment for testing.
    - It provides a clean debugger interface that will make it easier to fix bugs in your code.
    - It's free and little set-up required.

  - If you want to use an offline IDE, please make sure it conforms to the following requirements.
    - All projects you submit should be compilable with g++ 5.4.1 or higher. You may use any IDE that can utilize at least g++ 5.4.1.

      MinGW v8.1 (Windows equivalent to g++) can be downloaded from the following link: https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/8.1.0/threads-win32/sjlj/x86_64-8.1.0-release-win32-sjlj-rt_v6-rev0.7z/download

      Note the MinGW file must be opened and unpacked with 7-Zip (https://www.7-zip.org)

    - In class, the IDE I will be using is Code::Blocks. This is a free download for Windows. If you are using an offline IDE, you may use any IDE you choose as long as it supports the proper version of g++.
      http://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03-setup.exe
      - If you choose to use Code::Blocks, there is a document in the Course Resources section in eLearning to link the compiler to the IDE.

- **Note Taking Supplies** – I expect you to actively take notes. Anything we cover in class or you read in the textbook is fair game on a test. Have something available to collect all the knowledge given to you (be that in physical or digital form).

REQUIRED COURSE INFORMATION SECTION

Course Prerequisite: Prerequisite: CS 1336 with a grade of C or better or equivalent (placement test or AP credit).

Description of Course Content: 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.

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

- Ability to use single and multi-dimension arrays.
- Ability to implement linear and binary searches.
- Ability to implement simple sorting algorithms.
- Ability to implement structured data types.
- Ability to define and implement a class.
- Ability to use fundamentals of object-oriented design.

Departmental Attendance Policy: The Computer Science Department has implemented the following attendance policy:

If a student misses three consecutive classes, the student will receive a letter grade reduction to his or her final grade. This deduction is cumulative, so if a student misses three consecutive classes twice, the final grade will be reduced by two letter grades. **If a student misses four consecutive classes, the student will automatically receive an F for his or her final grade.**

Attendance will be taken each class through eLearning. Students will be presented with a question in class during the lecture and must answer the question in eLearning. **Asynchronous students must watch the lecture and complete the activity within 24 hours of the start of class for that day.**

WHAT I EXPECT OF EACH STUDENT

- **Ask for help at any time.** If you do not understand something or are having trouble implementing a concept, reach out. During lecture, before/after class, during office hours, at 3 AM while cramming for an exam, etc. I really mean any time. 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 a question.

- **Take responsibility for your education.** I will treat this course as similar to a professional setting as I can. I am not the type of teacher that lectures with slides and expects students to memorize them. I will teach by creating program examples in class. I will give challenging assignments to push you toward learning the intricacies of C++ and developing 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 learned on your own.

- **Practice time management skills.** All assignments (homework and projects) are designed to be worked on over a period of days or weeks. 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.**

- **Attend every class.** You are paying for an education. Don't waste your money by skipping class.

- **Make mistakes!** This is how you learn to do something. 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 the correct way to do it.

- **Read the chapter before the corresponding lecture** (see class schedule below)**.** I use class time to write programs that help illustrate the topics mentioned in each chapter. If a student doesn't have minimal knowledge of the concepts that will be covered for that chapter (which are gained by reading the chapter), it will be harder to get a deeper connection to what we are accomplishing in class.

WHAT EACH STUDENT SHOULD EXPECT

- **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.

- **Class commitment of 10 hours a week on average outside of class.** Students should be prepared to tackle multiple course-related activities each week (e.g. reading the textbook, studying for quizzes/exams, practicing programming, etc.). Procrastinating on an assignment will largely increase the number of hours spent each week.

- **Exams focused on application.** I do not create run-of-the-mill multiple choice exams that ask students to regurgitate things from memory. The exams are completely different than anything you have had in any other class (unless you've had me for a previous class). 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 as well (if not better) than me, and I'm going to push you toward that goal. 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.

THE INFORMATION YOU REALLY CARE ABOUT

Grading Scale:

98-100 A+	88-89 B+	78-79 C+	68-69 D+	Below 60 F
92-97 A	82-87 B	72-77 C	62-67 D	
90-91 A-	80-81 B-	70-71 C-	60-61 D	

Grade Components:

Projects (4)	40%
Exams (3)	30%
Preview Homework	10%
Review Homework	20%

General Grade Information: All grades will be available in eLearning. The Weighted Total column will give you the most accurate information concerning your grade. The weighted total is an approximation of your grade in the class based on the grades currently in eLearning.

I do not curve grades. Assignments are combined into categories so that a low grade for one item will not destroy your grade. There are also opportunities provided to help students who may have done poorly on projects and exams.

Grade Disputes: **All grade disputes must be reported within 1 week and resolved within 2 weeks of the grade in question being posted in eLearning.** Uncontested grades will become final after 1 week and cannot be disputed later. Announcements are made after each grade is posted so please check your grades promptly and reach out to the proper person.

I am responsible for grading your exams. If you have questions regarding your exam, please contact me through a private post on Piazza.

Everything else will be graded by a TA. Please address any grading concerns you have regarding these grades with the TA. **When you email the TA with questions about your grade, copy me on the email so that I am aware of the situation and can make sure it is resolved.**

Projects: Projects will be major programming assignments that supplement recently discussed topics and will be completed in two to three weeks. Projects are intended to take approximately 15-20 hours to complete overall; this includes the design, coding and testing process. Waiting until a couple of days before the due date to start the project is a bad idea. Not only does this introduce unnecessary stress into your life, it hardly ever ends well for the student. Most students score poorly on projects that are built in less than three days.

Projects will be divided into milestones as you would expect to see in the professional world. The milestones for each project are as follows:

- Design (4 days)
- Core implementation (7-10 days)
- Final implementation and testing (7-10 days)

Each milestone will be graded and the total of all three milestones will contribute toward the overall grade for the project. Each milestone will have a firm deadline and failure to meet the deadlines will have a negative impact on your grade.

Projects are individual endeavors and students are not to work in groups on any project. Students are permitted (and I openly encourage students) to discuss the project. Feel free to share ideas on the logic, but **DO NOT SHOW YOUR CODE TO OTHER STUDENTS.** When discussing logic, try to keep it general. If you give out every little piece of logic you have, there is a good chance the person you are helping will have very similar code as yours and may be flagged for being too similar. Be careful of posting your code online. Another student could use your code without your knowledge and could involve you in a code plagiarism referral.

Students should avoid using web sites like GitHub and Chegg for help on projects. Copying code from a web site is considered plagiarism and will be treated as such. If you find code on a web site, it is highly likely another student will find it as well which may cause both submissions to be flagged for similarity.

All projects will be submitted in ZyBooks and will be compared for originality. Any projects that are approximate or identical copies will be reported to the Office of Community Standards and Conduct, and I will accept their decision in regards to the grade if they believe that academic dishonesty has occurred.

Programming assignments will be graded on a 100 point basis. Not only will your project be graded on proper execution, but also things like efficiency, implementation and documentation. Keep in mind that you always want to write code that is easy to understand and is also easy to maintain. Fewer lines do not necessarily mean a better program. Please use comments liberally.

You are responsible for testing your project thoroughly before submission. I will not give you the exact test cases that will be used for grading before the project is due. As a computer scientist, you must be able to identify all possible input and make sure that your code produces proper output and does not crash.

Late Projects:

All project milestones will be due at 11:59 PM on the day listed in the project documentation. The final project code will be accepted up to 12 hours late with the following penalties

<= 1 hour	-5 points
1 – 3 hours	-10 points
3 – 12 hours	-25 points
12 – 24 hours	-50 points

Preview Homework: Preview homework assignments are questions based on the reading. These assignments will be in the form of quizzes in eLearning. Students will read the chapter and answer basic questions about the material to illustrate how well they understand the general concepts.

Review Homework: Review homework assignments are generally short coding assignments that can be done in 1-2 hours that measure how well you understand the material we have covered. These assignments are typically due 1 week from the date given.

Exams: Exams will cover chapters as listed below in the tentative course schedule. Exams will include a variety of question types including multiple choice, multiple answer and essay questions. Students are expected to be able to apply knowledge from all previous chapters in conjunction with the tested chapters. Exams are not created to make you feel smart; they are designed for you to demonstrate your understanding of the concepts. A high score on an exam exhibits a deep understanding of the topics.

An exam should not be missed except for the most extreme circumstances (such as hospitalization or death of an immediate family member). If you miss an exam, you must have documentation for the absence. A make-up exam may be given to students with valid documentation. The allowance of a make-up exam is at the sole discretion of the instructor.

All exams will be given online in eLearning. Students are expected to take the exam at a location of their choosing during the window of availability decided by the professor. The exams will be closed book and closed notes. **Each student is expected to take the test on their own without help from other people or internet sources.** All solutions to coding problems on the test will be submitted for similarity in an effort to maintain academic integrity.

You must start the exam within 15 minutes of the test start time. Failure to do so will result in a 10 point penalty on the exam.

Assignment Due Date Exceptions: In general, assignments are not accepted late except for the final project code. However, I know that life has a way of bringing the unexpected at the most inopportune times. If you have a life situation (personally or academically) that is creating difficulty for you to meet the given due date of an assignment, I will work with you to give an extension as long as you contact me **before** the due date.

ARE WE THERE YET?

All dates are subject to change at the discretion of the instructor

Date	Topic
1/19	Introduction to Debugging Intro to C
1/21	Character and C-string Functions
1/26	**Last day to add/swap classes** Characters, Strings and the String Class
1/28	Advanced File I/O
2/2	Advanced File I/O
2/3	**Last day to withdraw without "W"**
2/4	Recursion
2/9	Recursion
2/11	Recursion
2/23	Pointers
2/25	Pointers Smart pointers
2/27	**Exam 1 (Chapters 10, 12, 19)**
3/2	Smart pointers
3/4	Structures
3/9	Linked Lists
3/11	Linked Lists
3/15 – 3/21	**SPRING BREAK!**
3/23	Enumerated Data Types Pointers in C
3/25	Sorting Arrays
Virtual lecture	Searching Arrays
3/30	Quicksort
4/1	**Exam 2 (Linked Lists, C. 8, 9, 11)**
4/5	**Last day to withdraw**

4/6	Introduction to Classes
4/8	Introduction to Classes
4/13	Introduction to Classes Copy Constructor
4/15	Overloaded Operators
4/20	Overloaded Operators
4/22	Overloaded Operators
4/27	Static Members Inheritance
4/29	Inheritance
5/4	Polymorphism Virtual Functions
5/6	Polymorphism
5/13	**Exam 3 (C. 13, 14, 15)**

Assignment Calendar

All assignments due by 11:59 on the due date listed unless otherwise noted

All submissions are made using the provided links in eLearning

Assignment	Post Date	Due Date
C-Strings and String Library Preview	1/19	1/24
Review Homework 1	1/21	1/27
Advanced File Operations Preview	1/19	1/29
Project 1 Pseudocode	1/26	1/30
Recursion Preview	1/19	2/3
Review Homework 2	1/28	2/3
Project 1 Core Implementation	1/26	2/6
Project 1 Final Submission	1/26	2/12
Review Homework 3	2/4	2/14
Project 2 Pseudocode	2/23	2/26
Exam 1	2/27	2/27
Pointers Preview	1/19	3/1
~~Review Homework 4~~	Cancelled	
Project 2 Core Implementation	2/23	3/6
Structures and Enumeration Preview	1/19	3/8
Project 2 Final Submission	2/23	3/13
Pointers in C Preview	1/19	3/22
Project 3 Pseudocode	3/11	3/23
Review Homework 5	3/11	3/25
Searching and Sorting Preview	1/19	3/29

Review Homework 6	3/23	3/31
Exam 2	4/1	4/1
Project 3 Core Implementation	3/11	4/3
Objects and Classes Preview – Part I	1/19	4/8
Project 3 Final Submission	3/11	4/9
Project 4 Pseudocode	4/6	4/12
Review Homework 7	4/8	4/14
Objects and Classes Preview – Part II	1/19	4/18
Review Homework 8	4/15	4/23
Inheritance and Polymorphism Preview	1/19	4/26
Project 4 Final Submission	4/6	4/30
Review Homework 9	4/29	5/7
Exam 3	5/13	5/13

University Policies
For all other University policies, please visit http://go.utdallas.edu/syllabus-policies