COURSE INFORMATION:

Course Title: Programming Fundamentals **Section: 002** **Course Term: Fall 2024**
Meeting times: **MW 2:30pm - 4:45pm** **Classroom**: ECSS 2.305

COURSE MODALITY: IN PERSON

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.

NOTE :

- The Syllabus posted in the Course Book is just for basic info.
- Once the course starts, changes (if any) will be made only to the syllabus posted inside eLearning; changes would be highlighted with timestamps.

Professor / TA Contact Information:

Professor	**Priya Narayanasami** Contact: **via MS Teams ONLY** with **course name & section number**. (No emails) Office Hours: **M : 10am – 11am** **T : 4:30pm – 5:30pm** (Above listed hours inconvenient ? Reach out by Teams to set up an appointment) **Online office hours link : TBA**

NOTE :

- For messages sent on the weekend (after 5pm on Friday) , response will be sent on Monday.
- **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:

Will be provided later in eLearning.
You can also 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 PREREQUISITES AND COREQUISITES:

Prerequisite: None
Corequisite: None

COURSE DESCRIPTION:

CS 1436 Programming Fundamentals (4 semester credit hours) :

Introduces the fundamental concepts of structured programming. Topics include software development methodology, data types, control structures, functions, arrays, and the mechanics of running, testing, and debugging. Programming language of choice is C. The class is open to students in the School of Engineering and Computer Science only. Credit cannot be received for both courses, (CS 1336 and CS 1136) and CS 1436. Note that **a grade of C or better in this class is required to register for (CE 1337 or CS 1337).** (3-2) S

STUDENT LEARNING OBJECTIVES/OUTCOMES:

1. Ability to develop algorithmic solutions for use on computers.
2. Ability to perform console input and output, utilize basic operators, and perform sequential processing.
3. Ability to utilize the basic control structures for selection.
4. Ability to utilize the basic control structures for repetition logic.
5. Ability to perform sequential file input and output.
6. Ability to develop programs in a functional form.
7. Ability to process data in arrays.

REQUIRED TEXTBOOKS AND OTHER MATERIALS:

Required Textbook:

- Starting out with C++. From control structures through objects, Tenth Edition, by Tony Gaddis, Pearson Education, Inc. ISBN: 9780137450626.
- **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 ZyLabs :

We will be using an online platform for auto-grading of programming assignments and laboratory assignments.

Set up instructions:

1. **Go into eLearning for our course.**
2. **Navigate to Activities – Lecture portion.**
3. **Click on Class_01_zybook_setup.**
4. **Subscribe** (and pay $40)
    a. Use only your UTD email id (only then your grade can be linked with eLearning)
    b. **Do not go to the ZyBooks website directly and create a new account.**

Computer with Compiler:

- This is a programming course, and I believe in hands-on programming**.**
- **Please bring a laptop for completing in-class activities & laboratory assignments in class. For information about the Student Technology Initiative including information on financial aid and technology loaner program see:** https://oit.utdallas.edu/technologyinitiative/
- An online C++ compiler is available at: https://www.onlinegdb.com/online_c++_compiler

- Other suggested IDEs include:
  - CodeBlocks
  - Microsoft Visual Studio (Community) for Windows machines
  - Xcode for Macs
- Irrespective of the compiler you use for development, your grade will be based on passing the auto-graded tests run in ZyLabs.
  - You will not get credit for tests failed in ZyLabs, 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

---

TENTATIVE COURSE CALENDAR :

Class	Class Activity
08/19	First day of class - Syllabus Review and Intro to C++
Every Tuesday @ 11:59pm	Weekly homework due. Sometimes, multiple problems might be due – the student is responsible for checking the due dates. Hw1 due on 08/27 @ 11:59pm
09/02	*No class – Labor Day*
week of 09/16	Programming Challenges bundle#1 will be available. (Typically, 3 – 5 questions will be due – plan your schedule better).
Tuesday, 10/01 @ 11:59pm	Programming Challenges bundle#1 due. (**No late submissions on these**).
Wed, 10/02 **Review & Test 1**	We will meet in TI Auditorium **2:30pm – 3:15pm** Review concepts & if time permits, a few problems. **3:30pm – 4:45pm**: Test needs to be taken using your laptop.
Sat, 10/12	**Mid-term grades due**
week of 11/13	Programming Challenges bundle#2 will be available. (Typically, 3 – 5 questions will be due – plan your schedule better).
11/25 & 11/27	*No class – Thanksgiving break*
Tuesday, 12/03 @ 11:59pm	Programming Challenges bundle#2 due. (**No late submissions on these**).

Wed, 12/04 (most likely date) **Review & Test 2**	Location TBD
12/18	**Final grades**

TENTATIVE CONTENT COVERAGE SCHEDULE (STUDENTS ARE STRONGLY ENCOURAGED TO READ AHEAD)

Date	Lecture Material	Gaddis
08/19 - M	eLearning Access, Syllabus Review, Introduction to Computers & programming	Chapter 1
08/21, 08/26, 08/28	Introduction to C++ & Expressions and Interactivity	Chapters 2 & 3
09/04 – 09/11	Decision Structures	Chapter 4
09/16 – 09/30	Loops	Chapter 5
10/02 – W	Revision and Test 1 (see schedule for details)	
10/07 & 10/09	Files	Chapter 5
10/14 – 10/21	Functions – intro	Chapter 6
10/23, **10/28**, 10/30, **11/04** 11/06,**11/11**	Arrays – intro & immersion	Chapter 7
11/13, 11/18, 11/20,12/02	Work on problems involving all the concepts covered until now.	
11/25 & 11/27	*No class – Thanksgiving break*	
12/04 – Wed	Revision and Test 2 (see schedule for details)	

The instructor reserves the right to modify this calendar as they deem necessary.
Please see eLearning for announcements regarding changes to the calendar.

INSTRUCTOR POLICIES:

Grading Policy:

Your course average will be calculated as follows:

- Test/Exam #1 – 30%
- Test/Exam #2 – 30%
- Homework – 15%
- Attendance/In-class activities/Participation Exercises – 15%
- Laboratory Assignments – 10%

Exams / Tests :

- **There will be 2 tests (equal weight).**
- Please check the tentative schedule for the date/time and location.
- Students are expected to take it using their laptop using Respondus Lockdown browser, and it will be deployed in the **eLearning\Test** tab.
- Further instructions will be provided a week before test 1.
- **Closed book and closed notes – cheat sheets are <u>not</u> allowed.**
- You will be required to code in the tests.

- To avoid the need for memorization as much as possible, necessary documentation or sample code will be provided.
- Make-up tests :
    - 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 test will <u>not</u> be given.**
    - There will be 20% penalty for makeup-exam requests citing non-emergency purposes.

Homework or Programming Assignments:

- Weekly Assignment(s) will be added under **eLearning/ Homework or Programming assignments** tab.
- They need to be accessed via eLearning only; that link will take the students into ZyBooks where the student can read about the problem in detail; the student may choose to code in any IDE ; the student needs to paste the solution into ZyBooks and click on **Submit for grading** button in ZyBooks.
- Each assignment will include a due date and time.
- Late assignments will not be accepted; **excuses will NOT be entertained either.**
- **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.
    - You should only ask for help with debugging as a last resort.
    - **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.*
- Historically, the students who skip programming assignments, 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.

In-class activities / Participation activities / Lecture portion activities / Attendance:

- Typically, after a day's concept is covered, students will be given at least 3 problems to work on during lecture portion.
    - Students are expected to work on all the problems that have been given.
    - Towards the end of lecture time, students will be asked to submit one of them (Professor's choice) as in-class-activity submission.
    - Students are expected to submit their code for that problem in a link posted under **eLearning/Activities-Lecture** portion.
    - This is **submission grade activity** – meaning that the program need not work perfectly; as long as there is reasonable effort, points will be given.
- Rarely, roll call might also be done.
- 3 low scores for in-class activities will be dropped.
- **There is no make-up for missed attendance/exercise.**
- **No excuses are necessary or desired.**
- If you have an extended illness or other crisis that will cause you to miss more than two classes, reach out to the professor via Teams one-on-one chat for guidance on how best to proceed.

Laboratory (Lab) Assignments:

- After the day's concept is covered, one assignment will be uploaded under **the eLearning/ Activities - Lab portion** tab.
- Students are expected to finish those problems during the time allocated for the lab problems.
- Students will be given time in course meetings and assistance with the laboratory assignments.
- **You may not send your source code to the lab assistant or the instructor unsolicited by Teams expecting us to debug/fix it.**
- Before seeking help from the lab assistant, you should have narrowed down the problem. *When the lab assistant looks at your code, they will expect to see debug statements, to see that you have done your best to locate the error(s). If your code does not include debug statements or is not neatly organized with indentation, meaningful variable names, etc., the lab manager will ask you to check back after you have corrected your style and added debug statements.*
- For early assignments, it is likely you will have ample time to complete the assignment in the meeting as long as you are keeping up with your coursework, including assigned reading and lectures, and use your time in the meetings wisely.
- **Late submissions will not be accepted for any reason. Please do not send excuses.**

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		

CS/CE 1337, TE 1337, CS 1325, etc. have a minimum grade requirement for CS 1436. Make sure you know the grade you need in CS 1436 to be able to sign up for your next programming course.

Questions about your grade ?

- **Homework assignment, in-class activities, laboratory assignments, problems in programming challenge bundles :**
  - The TA will go through your code to see if the style guidelines and logic are present in your code; they will take off points if the submitted code does not comply with style guidelines or if they find that you have written code to satisfy the test cases rather than have logic in your code. So, in these situations, even though your test cases might pass, and your initial score was 100, your score may come down.
  - It is the responsibility of the students to keep up with their scores.
  - The **TA grades these; you need to notify both the TA and the instructor using Teams within 3 days** after the date the grade is posted in the grade book on eLearning.
- **Test / exam :**
  - The Professor will be grading the tests **;** so, you **need to 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.
- Your request for 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, disregarded the 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.

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.
- **ZyBooks 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

Class Rules:

- **Do not be late to the class**.
- **Reading ahead is expected.**
- **Active participation is expected**.

Do not use laptops / mobile phones for non-academic purposes of this course - students may be asked to leave the class and their attendance will be voided for that day.

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, distracted due to usage of cellphone, 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 small changes. Note how the changes affect the program translation and/or execution.
  - Pick a few of the Programming Exercises at the end of the chapter or in ZyBooks 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 textbook.

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.

Class Recordings

The instructor will record the lecture portion of the class – if needed, the recordings are to be used for revisiting a concept that was taught in class and in case anyone was absent, they have access to the covered material.

Students will not be permitted to attend class online.

Students are expected to follow appropriate University policies and maintain the security of passwords used to access recorded lectures. Students are expressly prohibited from recording any part of this course. Recordings may not be published, reproduced, or shared with those not in the 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.

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.