COURSE INFORMATION:

Course Title: Programming Fundamentals
Section: 002 (MW 2:30pm – 4:45pm)
Course Term: Spring 2024

INSTRUCTOR CONTACT INFORMATION:

Name: Laurie Thompson
Office: ECSS 3.701
Telephone Number: (972) 883-6326 – Email is preferred, you may leave a voicemail here in case of emergency.
Email Address: **Emails must be sent from your UTD account with a subject that begins with: CS1436.002.** It is likely that emails sent on the weekend (after 5pm on Friday) may not be responded to until Monday.
Laurie.Thompson@utdallas.edu

Office Hours: Mondays: 1:00pm – 2:00pm
Wednesdays: 1:00pm – 2:00pm
Thursdays: 1:00pm – 3:00pm
Other times by appointment.

Do not procrastinate. Historically, my office hours become significantly busier as we near a due date. I WILL NOT schedule meetings to assist students with work that is due within the next 24 hours.

Poor planning on your part does not necessitate an emergency on mine. – Bob Carter

---

GRADER CONTACT INFORMATION:

Will be provided later in eLearning.

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. Lab fee of $30 is required. 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 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

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. Check the course Announcements and your UTD email account for updates. Online meetings will be recorded and posted in MS Streams.

REQUIRED TEXTBOOKS AND MATERIALS:

Required Textbook:

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

Editions from the seventh or newer are also acceptable except Revel.

See the TENTATIVE COURSE CALENDAR later in the document for reading assignments.

Required Subscription to Zylabs:

We will be using an online platform for interactive practice, feedback, and auto-grading.

1. Click on the Zylabs link in eLearning **(Do not browse directly to the ZyBooks website, start from the link here in eLearning!)**
2. Subscribe (and pay) Use your UTD email.

A subscription is $40. Students may begin subscribing on Jan 02, 2024 and the cutoff to subscribe is Apr 26, 2024. Subscriptions will last until May 24, 2024.

Computer with Compiler:

Use of a laptop for completing laboratory assignments and other work 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:

Microsoft Visual Studio (Community) for Windows machines
Xcode for Macs

Whichever compiler you use for development, your grade is based on passing the auto-graded tests run in Zylabs. Zylabs uses a Linux based compiler. Minor adjustments may be needed when porting from your windows or mac-based compiler.

You will not get credit for tests failed in Zylabs, regardless of the results on your machine.

CLASS MATERIALS:

Slides, Sample Programs, etc.:

Other materials including the syllabus, slides, sample programs, assignments, and laboratory exercises etc. will be posted and or linked in eLearning.

https://elearning.utdallas.edu

TENTATIVE COURSE CALENDAR

Date	Lecture Material	Preparation Due (Reading and/or Recording)	Laboratory Exercises – These are due at 11:59pm, the first Sunday after the meeting
January 17	eLearning Access, Syllabus Review, Introduction to Computers	Chapter 1 text	
January 22	Number Systems & Introduction to C++	Number Systems Demo & Chapter 2 text	
January 24	Introduction to C++		
January 29	Introduction to C++ & Zylabs Introduction		Zylabs Practice
January 31 (Census Day)	Expressions and Interactivity	Chapter 3 text	Laboratory exercise
February 5	Expressions and Interactivity		Laboratory exercise
February 7	Expressions and Interactivity		Laboratory exercise
February 12	Expressions and Interactivity		Laboratory exercise
February 14	Expressions and Interactivity		Laboratory exercise
February 19	Expressions and Interactivity		Laboratory exercise
February 21	Making Decisions	Chapter 4 text	Laboratory exercise
February 26	Making Decisions		Laboratory exercise
February 28	Making Decisions		Laboratory exercise
March 4	Making Decisions		Extra instruction, review, practice, or lab exercise
March 6 (2:00pm – 5:30pm)	**Exam #1 (120 minutes)**		
March 11	**Spring Break – No Lecture**		
March 13	**Spring Break – No Lecture**		
March 18	Functions	Chapter 6 text	Laboratory exercise
March 20	Functions		Laboratory exercise
March 25	Functions		Laboratory exercise
March 27	Functions		Laboratory exercise
April 1	Loops	Chapter 5 text	Laboratory exercise
April 3	Loops		Laboratory exercise
April 8	Loops		Laboratory exercise
April 10	Loops		Laboratory exercise
April 15	Loops & Files		Laboratory exercise
April 17	Files		Laboratory exercise
April 22	Arrays	Chapter 7 text through Section 10	Laboratory exercise
April 24	Arrays		Laboratory exercise
April 29	Arrays		Laboratory exercise
May 1	Arrays		Extra instruction, review, practice, or lab exercise
May 8 (1:30pm – 6:00pm)	**Exam #2 (150 minutes)**		

The instructor reserves the right to modify this calendar, as necessary. Please see eLearning for discussions/announcements regarding changes to the calendar.

GRADING POLICY:

Your course average will be calculated as follows:

Exam #1 – 35%
Exam #2 – 35%
Programming Assignments – 15%
Laboratory Exercises – 10%
Attendance/Participation – 5%

The instructor intends to assign letter grades as shown below.

Averages	Letter grade
≥ 96.5	A+
≥ 92.5 AND < 96.5	A
≥ 89.5 AND < 92.5	A-
≥ 86.5 AND < 89.5	B+
≥ 82.5 AND < 86.5	B
≥ 79.5 AND < 82.5	B-
≥ 76.5 AND < 79.5	C+
≥ 72.5 AND < 76.5	C
≥ 69.5 AND < 72.5	C-
≥ 66.5 AND < 69.5	D+
≥ 62.5 AND < 66.5	D
≥ 59.5 AND < 62.5	D-
< 59.5	F

Grading Concerns:

If you think there is a mistake in the grading of your programming assignment or laboratory exercise, **you must notify both the grader and the instructor (email the grader and copy the instructor)** of this by email **within two weeks** after the date the grade is posted in the grade book on eLearning. 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 specific constructs or functions. To get the maximum credit you MUST read the directions carefully and test your programs thoroughly.

If you think there is a mistake in the grading of your exam and would like to request that it be regraded, **you must notify the instructor** of this by email **within two weeks** 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.

COURSE & INSTRUCTOR POLICIES:

Exams:

Course Exams will be administered in the testing center.

Make-up examinations will be administered **only for well-documented emergencies**. A student must make every attempt possible, via telephone and email, to notify the instructor that he/she will miss a scheduled exam **before** the scheduled date and time or **immediately** thereafter. **If notification is not received in a timely manner, no make-up will be given.**

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.

Programming Assignments:

All assignments will be announced and submitted using eLearning/Zylabs. You will be given at least one week to complete each assignment. Each assignment will include a due date and time. **Late submissions will not be accepted for any reason. Please do not send excuses.**

Laboratory Exercises:

Exercises will be announced and submitted using eLearning/Zylabs. Laboratory exercises are typically due the Sunday night following the laboratory meeting at 11:59pm. **Late submissions will not be accepted for any reason. Please do not send excuses.**

You will be given time in course meetings and assistance with the laboratory assignments. For early assignments, it is likely you will have ample time to complete the assignment in the meeting if you are keeping up with your coursework, including assigned reading and lectures, and use your time in the meetings wisely.

Your two lowest laboratory exercise scores will be dropped from the calculation of your grade at the end of the semester.

Help with Assignments & Exercises Outside of Class Meetings:

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 and helps you find errors. This includes the use of debug statements in your code. You should also learn to use an integrated debugger. You should only ask for help with debugging as a last resort.

The instructor or grader will help you find errors during office hours, but you should have narrowed down the problem before asking for help. *When we look at your code, we will expect to see debug statements, to see that you have done your best to locate the error(s). If we do not see these, we will ask you to check back after you have added them.*

You MAY NOT send your source code to the grader or instructor unsolicited by email expecting us to debug/fix it. This is not reasonable. There are too many of you for us to do this.

Tutoring:

The CSMC (Computer Science Mentor Center) also provides walk-in help/tutoring. See the course home page in eLearning for a link to the CSMC website.

Academic Integrity:

All assignments, exercises, and exams are to be individual efforts. You are not to collaborate with other students. Prior to the assignment/exercise/exam due date, you are not to: discuss solutions with other students in anything but the most general terms (exams you may not discuss at all), distribute your code to others, or publish your code. Copying programming assignments exercises, in whole or in part, from other students will be considered an act of scholastic dishonesty. Copying assignments/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 submitted in past semesters, or code from the Internet. Plagiarism detection software will be employed to detect copying of code. **Zylabs 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.

Participation/Attendance:

Attendance will be taken to determine attendance and active participation in course meetings. This may be accomplished with small attendance exercises or with attendance checks at any or multiple points during meeting time. You will be considered absent if you miss any of these checks. **There are no make-ups for missed attendance/exercise.**

After three absences your attendance score will be reduced on a prorated basis.
No excuses are necessary or desired. If you miss more than three absences, your attendance score will be reduced regardless of reason.

Please do let the instructor know if you have an extended illness or other crisis that will cause you to miss more than two meetings, so that I can coordinate with you and advise you on how best to proceed.

You will be excused from meetings early only if you show the instructor/grader that you have completed the laboratory exercise(s) and received full-score for the automated tests. If you leave before you are excused, you will be marked absent.

Extra Credit Work:

Extra credit work will not be given to *individual* students.

STUDENT SUCCESS & RESPONSIBILITIES:

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 in the slides and slide recordings.
- You are responsible for all material discussed in course meetings.
- You are responsible for all material supplied on eLearning (including announcements and discussion postings).
- Students are expected to be respectful of each other and of the course instructor. Disruptive behavior will not be tolerated.

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 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 approximately **12** hours per week **outside of scheduled meetings** for reading, watching recordings, 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.
- 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, the course is constantly moving forward.
- If you do not understand a concept: Reread the text, review the slides, or recordings. Arrange a meeting with your instructor and come to the meeting with a list of specific questions. Visit the Computer Science Mentor Center (this can be done virtually). Again, come 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. 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). This course is like a mathematics course – you need to work **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 Zylabs and write programs that satisfy the requirements. This is good practice for the types of coding questions I will ask you on the exam.
- 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 AccessAbility Resource Center accommodation. Failure to comply with these University requirements is a violation of the Student Code of Conduct.

Class Recordings

Students are expected to follow appropriate University policies and maintain the security of passwords used to access recorded lectures. Unless the AccessAbility Resource Center has approved the student to record the instruction, 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 AccessAbility Resource Center accommodation. Failure to comply with these University requirements is a violation of the Student Code of Conduct.

NOTE: if the instructor records any part of the course, then the instructor will need to add the following syllabus statement:

The instructor may record meetings of this course. These recordings will be made available to all students registered for this class if the intent is to supplement the classroom experience. If the instructor or a UTD school/department/office plans any other uses for the recordings, consent of the students identifiable in the recordings is required prior to such use unless an exception is allowed by law.

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.
