

**COURSE INFORMATION:**

**Course Title:** Programming Fundamentals  
**Section:** 001  
**Course Term:** Fall 2024

---

**INSTRUCTOR CONTACT INFORMATION:**

**Name:** Brian Ricks  
**Office Location:** ECSS 4.701  
**Office Number:** 972-883-2674  
**Email Address:** absolutefunk@utdallas.edu  
Emails must be sent from your UTD account with a subject that begins with: CS1436.001  
It is likely that emails sent on the weekend (after 5pm on Friday) may not be responded to until Monday.

**Office Hours:** Tuesday: 11:30am – 1pm  
Wednesday: 2pm – 3:30pm

**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*

*If you ask me for help early in the assignment cycle, I typically have more time to spend with you.*

---

**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. 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
- 

## **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 Teams.

---

## **COMPUTER SCIENCE MENTORING CENTER (CSMC)**

The Computer Science Mentoring Center (CSMC) is a free resource available to all students taking this class. The CSMC provides assistance in many areas including:

- Understanding core concepts related to this class
- Developing a logical framework for a program
- Connecting programming constructs to the logic of the program
- Assisting in solving syntax and logical errors in your code
- Exam reviews and reworks (by faculty request)

The mentors will meet with you 1-on-1 to address your specific problem areas. Their goal is to help you understand what is wrong and how to fix it, but they will not do the work for you. For more information about the CSMC, including location and hours of operation, please visit <http://csmc.utdallas.edu>

---

## 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.

**Any version Seventh or newer is also acceptable.**

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

### Required Subscription to zyLabs:

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

1. Click on a zyLabs link in eLearning (**Do not browse directly to the zyBooks website, start from any zyLabs link in eLearning**)
2. Subscribe (and pay). Use your UTD email. Be sure to put **001** as the section number.

### 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++ IDE, OnlineGDB, which we will use in class is available at:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

Whichever compiler you use for development, your grade is based in-part 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. If writing code in OnlineGDB, its compiler is also Linux based.

**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, quizzes, assignments, and participation exercises etc. will be posted and or linked in eLearning.

<https://elearning.utdallas.edu>

---

## TENTATIVE COURSE CALENDAR:

Date	Lecture Material	Gaddis	Laboratory Assignment
August 19	eLearning Access, Syllabus Review, Introduction to Computers	Chapter 1	
August 21	Number Systems & Introduction to C++	Chapter 2	
August 26	Introduction to C++		
August 28	Introduction to C++		
<b>September 2</b>	<b>Labor Day – No Lecture</b>		
September 4	Expressions and Interactivity & zyLabs Introduction	Chapter 3	zyLabs Practice
September 9	Expressions and Interactivity		Lab #1 – Arithmetic Operators
September 11	Expressions and Interactivity		Lab #2 – Math Functions
September 16	Expressions and Interactivity		Lab #3 – Mathematical Calculations and Functions
September 18	Expressions and Interactivity		Lab #4 – Mathematical Calculations and Output Formatting
September 23	Expressions and Interactivity		Lab #5 – Mathematical Calculations and Output Formatting
September 25	Expressions and Interactivity		Lab #6 – String Reads and Concatenation
<b>September 26-27</b>	<b>Exam #1 in the Testing Center</b>		
September 30	Making Decisions	Chapter 4	Lab #7 – String Functions <i>length</i> and <i>at</i>
October 2	Making Decisions		Lab #8 – Decisions
October 7	Making Decisions		Lab #9 – Decisions
October 9	Making Decisions		Lab #10 – Decisions
October 14	Functions	Chapter 6	Lab #11 – Decisions
October 16	Functions		Lab #12 – Functions w/o Loops
October 21	Functions		Lab #13 – Functions w/o Loops
October 23	Functions		Lab #14 – Functions w/o Loops, pass by reference
October 28	Loops	Chapter 5	Lab #15 – while Loop
October 30	Loops		Lab #16 – do-while loop
<b>October 31-November 1</b>	<b>Exam #2 in the Testing Center</b>		
November 4	Loops		Lab #17 – for Loop
November 6	Loops & Files		Lab #18 – Loops & Files
November 11	Loops & Files		Lab #19 – Loops & Files
November 13	Arrays	Chapter 7 through Section 10	Lab #20 – Arrays
November 18	Arrays		Lab #21 – Files & Arrays
November 20	Arrays		Lab #22 – Two-Dimensional Arrays & Random Numbers
<b>November 25</b>	<b>Thanksgiving Week – No Lecture</b>		
<b>November 27</b>	<b>Thanksgiving Week – No Lecture</b>		
December 2	Arrays		Open lab: practice exercise, tutoring, makeup lecture
December 4	Review Session		Open lab: practice exercise, tutoring, makeup lecture
<b>December 9-11</b>	<b>Exam #3 in the Testing Center</b>		

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

## INSTRUCTOR POLICIES:

### Grading Policy:

Your course average will be calculated as follows:

Exam #1 – 20%

Exam #2 – 25%

Exam #3 – 25%

Programming Assignments – 18%

Laboratory Assignments – 10%

Quizzes – 2%

The instructor intends to assign letter grades as shown below. **An average with a fractional portion of five tenths or above will be rounded up to the next whole number for determining the letter grade.**

Averages	Letter grade
97+	A+
93-96	A
90-92	A-
87- 89	B+
83-86	B
80-82	B-
77-79	C+
73-76	C
70-72	C-
67-69	D+
63-66	D
60-62	D-
Below 60	F

**Some courses (For example CS/CE 1337) have a minimum grade requirement for CS 1436. Make sure you know what grade you need in CS 1436 to matriculate to your next programming course.**

### Grading Concerns:

If you think there is a mistake in the grading of your assignment, **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 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 quiz or 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.

### 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.

### Exams:

Course Exams will be administered in the testing center. As there are multiple sections of this course, **make sure to register for the exams for this section.**

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

**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.**

### Course 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. You may submit an assignment up to 24 hours after the due date and time with a 10-point penalty. **No excuses will be entertained for late assignments.**

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

The instructor will drop your lowest assignment grade **if** you submit all assignments as scheduled and receive a grade of 60 or higher on each of the assignments. Historically, 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, and tend not to perform well on exam questions testing the material covered by the assignment.

## Laboratory Assignments:

Assignments will be announced and submitted using eLearning links to zyLabs. **Late submissions will not be accepted for any reason. Please do not send excuses.**

You will be given time in lab 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 as long as you are keeping up with your coursework, including assigned reading and lectures, and use your time in the meetings wisely.

**You may not send your source code to the lab assistant or the 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. 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.

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.*

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

## Quizzes:

There will be at least one quiz given during the semester, to test foundational knowledge of a specific topic of importance to programming. All quizzes will be given on eLearning, with a single attempt (once started, the quiz must be submitted in a single sitting). No make-up for quizzes will be provided – you will have at least one week to complete each quiz.

## 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/exam due date (this includes any late window), you are not to: discuss solutions with other students in anything but the most general terms (for quizzes/exams you may not discuss at all), distribute your code to others, or publish your code. Copying of programming assignments/quizzes, in whole or in part, from other students will be considered an act of scholastic dishonesty. Copying of assignments/quizzes 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/quizzes 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.

### Extra Credit Work:

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

### 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 **10 -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 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 2021 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 group 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**

Students are expected to follow appropriate University policies and maintain the security of passwords used to access recorded lectures. Unless the Office of Student AccessAbility 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 Office of Student AccessAbility accommodation. Failure to comply with these University requirements is a violation of the [Student Code of Conduct](#).

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.

---

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