CS 1325 Course Syllabus

COURSE INFORMATION:

Course Number: CS 1325.501
Course Title: Introduction to Programming Fundamentals
Course Term: Spring 2023
Schedule: M W 5.30 – 6:45 PM
Location: ECSS 2.305

INSTRUCTOR CONTACT INFORMATION:

Email Address: Srimathi.Srinivasan@utdallas.edu
Office Location: ECSN 2.924
Office Hours: Tuesday 1 – 2.30 PM in MS Teams
                    Wednesdays 4 – 5 PM in person
                    Click here to enter my Office room

GRADER CONTACT INFORMATION:

TA: TBA
TA email:
Office hours:
Location:
Email: When you send a mail to me or TA, please **specify 1325.003 in the subject of the email.**

COURSE PREREQUISITES AND COREQUISITES:

CS 1336 or equivalent

COURSE DESCRIPTION:

Computer programming in a high level, block structured language. Basic data types and variables, memory usage, control structures, functions/procedures and parameter passing, recursion, input/output. Programming projects related to engineering applications, numerical methods. This class is designed for Electrical and Mechanical Engineering majors and cannot be used to fulfill Major Requirements for Computer Engineering, Computer Science, Software Engineering, Telecommunications Engineering majors.

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 Accessibility accommodation. Failure to comply with these University requirements is a violation of the Student Code of Conduct.

STUDENT LEARNING OBJECTIVES/OUTCOMES:

1. Ability to use fundamental programming constructs: assignments, loops, conditions.
2. Ability to process data in arrays.
3. Ability to develop programs in a functional form.
4. Ability to perform sequential file input and output.
5. Ability to express algorithms that solve elementary engineering and scientific problems

---

REQUIRED TEXTBOOKS AND MATERIALS:

"C: How to Program", Deitel and Deitel, 8th Edition, Pearson, 2016. ISBN-13: 978-013-397689-2/ISBN-10: 0-13-397689-0

Suggested Materials

"C Programming for Absolute Beginners" (2nd Edition) by Michael Vine. Course Technology, 2009
"Engineering Problem Solving with C", Etter, Pearson, 2013 (0-13-608531-8)
"C for Engineers and Scientists", Cheng, McGraw Hill, 2010 (978-0-07337605-9)

C/C++ Compiler:

Any standard C/C++ compiler and Integrated Development Environment (IDE) can be used to develop, debug and run your programs. Microsoft Visual Studio, Microsoft Visual Express, Code::Blocks, NetBeans, Eclipse and jGRASP are a few popular tools.

I will use Microsoft visual studio for class demonstrations

Slides, Sample Programs, etc.:

Other materials including the syllabus, slides, and sample programs will be shared though UTD box folder.

Important Dates and Times	• First day of class: Wednesday, January 18th, 2022 • Exam 1: Wednesday, March 8th, 2023 @ Testing center • Exam 2: Wednesday, May 3rd, 2023 @ Testing center

Please register for the exams at UTD testing center as early as possible.
Testing Center - The University of Texas at Dallas (utdallas.edu)

Note: Your classes on 18th will be an online class and *Click here to join the meeting*
 Your class on January 23rd is cancelled. Classes will be in person from January 25th

Tentative Course Calendar

Week	Dates	Class Activity/Notes	Read …
1	Jan 18th	**Review of syllabus – Online class**	Chapter 1
2	Jan 23rd, 25th	**Classes cancelled on 23rd** Number Systems, C program	Chapter 2
3	Jan 30th, Feb 1st	Operators	Chapter 3
4	Feb 6th, 8th	Selection Statements, Flow Control	Chapter 4
5	Feb 13th, 15th	Functions, Recursion	
6	Feb 20th, 22nd	Arrays – Functions and Parallel	Chapter 5
7	Feb 27th, Mar 1st	Arrays – Search and Sort	
8	**March 6th, 8th**	2D Arrays; **Exam 1: Wednesday, March 8th**	**5.30 – 7.00pm (90 mins)**
9	**March 13th, 15th**	**Spring Break**	
10	March 20th, 22nd	C-Strings and Pointers	Chapter 6
11	March 27th, 29th	C-Strings Operations	
12	April 3rd, 5th	Files	Chapter 7
13	April 10th, 12th	Arrays and Files,	
14	April 17th, 19th	Structures, malloc and Files	
15	April 24th, 26th	Structures and Files	
16	**May 1st, 3rd**	Review; **Exam 2: Wednesday, May 3rd**	**5.30 – 7.00pm (90 mins)**

Weekly Assignments will be assigned on Wednesdays and are due by next Tuesday midnight.
Each weekly assignment will require the students to spend a few hours to even days programming in a computer. Right way to approach any programming assignment is to start right away & ask for help when you get stuck (you can approach the instructor or TA for help). Do not waste several hours trying to fix a small glitch. In simple words, your approach will determine whether the programming assignments provide an enjoyable learning experience or end up as painful & seemingly useless activities.

Assignments will be graded on a 100-point basis, utilizing the following criteria:

		Max Score
Pseudocode	High level explanation of steps at the top of program/each function	10%
Source Code	Overall design of the program	40%
Source Code	Assigning meaningful names for variables and functions	10%
	Formatting and coding style	10%
Execution	Test cases	30%
Total		**100%**

GRADING POLICY:

Grading Criteria	In class exercises: 10%, %, Assignments: 40%, Exam 1: 20%, Exam 2: 25%, Attendance 5% Historically students who skip programming assignments, or 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.
Programming Assignments	Programming assignments are given every week. **on Wednesdays and are due by next Tuesday midnight.** Programming assignments must be submitted through eLearning. You need to submit only .c files for individual assignments, unless explicitly stated otherwise.
In class Exercises	Exercises may be given in lecture with or without previous notification. **There are no make-ups for these. It must be submitted by next day 11.59PM of the lecture day.**
Make-up Exams	Make-up examinations will be administered **only for well-documented emergencies**. A student must make every attempt possible, via 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.**
Extra Credit	Practice problems may be assigned to get some extra credits.
Late Work	**Late submissions will not be accepted** for Programming assignments or In class activities as the solutions will be discussed in class the next day.
Class Attendance	Regular attendance is highly recommended. As per the Department of Computer Science policy, three consecutive absences lead to one letter grade drop. Four consecutive absences lead to a F. http://cs.utdallas.edu/education/undergraduate/attendance-policy/
Classroom Citizenship	The instructor encourages students to take active part in class discussions. No question is too simple/stupid to be asked. So, do not hesitate. Use of Laptops and smart phones for purposes other than related to class work is strictly prohibited. Sometimes, I may insist that laptops be closed.
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 go to http://go.utdallas.edu/syllabus-policies for these policies.

Letter grades will be assigned as follows:

97-100	A+	94-97	A	90-94	A-
87-90	B+	84-87	B	80-84	B-
77-80	C+	74-77	C	70-74	C-
67-70	D+	64-67	D	60-64	D-
Below 60	F				

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

- Read your assigned reading <u>before</u> the lecture before course meetings. You are expected to have an understanding of the assigned textbook material before meetings.
- Attend every meeting and pay close attention.
- Dedicate 9-10 hours per week <u>outside of class meetings</u> to CS 1325 for reading, practicing & writing code, assignments, exercises and studying for exams.
- 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).
- Complete the Checkpoint questions at the end of the sections of the text.
- Complete the Review Questions and Exercises at the end of the chapters of the text.
- Pick a few of the Programming Exercises at the end of the chapter and write programs that satisfy the requirements given. This is good practice for the types of coding questions I will ask you on the exam.
- Start your assignment immediately. All assignments 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.

- Ask for help!
  - The instructor is available to help during office hours.
  - Be proactive.
    - Don't wait till the day an assignment is due to seek help. Please note that I do not have office hours every day. If you wait till close to the assignment due date to seek my help, it is possible that I won't have any office hours that day or there may be many students who will be competing for my time when you come to my office hours. **You may not email your code to the instructor or grader expecting us to find your errors.**
    - You can also ask help at CSMC https://csmc.utdallas.edu/
    - Don't wait till the end of the semester to seek help. If you have gotten far behind in your coursework or have done significant damage to your course average, I may not be able to help.

COURSE & INSTRUCTOR POLICIES:

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.

You may not send your source code to the grader or instructor unsolicited by email expecting us to debug it. 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. We will help you find errors in person during office hours, but you should have narrowed down the problem before coming to see us.

Academic Integrity:

All assignments, exercises and exams are to be individual efforts. You are not to collaborate with other students. Prior to the assignment due date, you are not to: discuss assignment solutions with other students, distribute your code to others, or publish your code. Copying of programming assignments, exercises or exams, in whole or in part, from other students will be considered an act of scholastic dishonesty. Copying of assignments from previous semesters will be considered an act of scholastic dishonesty.

For programming assignments, you may use source code provided by the instructor. You are not to view, copy, or distribute code from any other sources, including code from other students, code from assignments submitted in past semesters, or code from the Internet. Plagiarism detection software will be employed to detect copying of code.

Grading Concerns:

If you think there is a mistake in the grading of your assignment or exercise and would like to request that it be regraded, **you must notify both the grader and the instructor (email the grader and copy the instructor)** of this by email within **one week** 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, 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 particular 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 regrades 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.

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."

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 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.