

Texas Tech University
Dept. of Computer Science

Course Name: Programming Principles II - (C)
Section Number: 001

Course Number: CS1412
Semester: Fall 2016

Instructor: Cong Pu (Ph.D., Instructor)
Email: cong.pu@ttu.edu

Office: ENGCTR 304
Office Hours: 10:00AM-11:00AM (WF)

Class Room: ENGCTR 110
Lab: Mechanical Engineering 219

Class Hours: 9:00AM-9:50AM (MWF)
Lab Hours: 11:00AM-1:50PM (W)

Lab TA: Akalanka Mailewa
Lab TA Email: akalanka.mailewa@ttu.edu

Lab TA Office: ENGCTR 201A
Lab TA Office Hours: 2:00PM-4:00PM (W)

Catalogue Listing: Advanced procedural programming. Topics include recursive functions, parameter passing, structures, records, memory allocation, exception handling, and abstract data types.

Textbook (required): C Programming: A Modern Approach, 2nd edition, K. N. King. W.W. Norton & Company, April 2008, ISBN-10: 0393979504, ISBN-13: 978-0393979503.

Course Objectives:

The objective of this course is to introduce advanced constructs in C programming language. Students will apply these constructs and problem solving methodology to complex problems.

Key Topics:

1. Review problem-solving methodology and basic constructs.
2. Advanced data types and structures:
 - a. Structs, records.
 - b. Pointers, linked lists.
 - c. Dynamic memory allocation and de-allocation.
 - d. Abstract data types (ADT).
3. Functions (recursive, parameter passing).
4. Programming practices:
 - a. Standard library usage.
 - b. Unix shell usage.
 - c. Exception handling.

Course Prerequisites: CS1411 (Programming Principles I) or ECE 1304.

Expected prior knowledge and skills in: basic problem-solving methodology and ability to program in some high-level language. Students are expected to have prior knowledge of: arrays, variables, assignment statements, operator precedence, control statements (e.g., if-else), loops (e.g., while and for) and functions.

Learning Outcomes & Assessment Methods:

Students who have completed this course should have the ability to:

Objective	ABET Outcomes	Assessment Methods
1. Capable of applying learned methodology to solve advanced problems.	a, b, c	Various lab, assignments, projects, all exams, and quiz.
2. Comprehend and apply advanced data types and structures.	a, b, c	Various lab, assignments, projects, all exams and quiz.
3. Comprehend and apply parameter passing.	a, b, c	Various lab, assignments, all exams, projects and quiz.
4. Comprehend and apply recursion.	a, b, c	Various lab, assignments, all exams, projects and quiz.
5. Comprehend and apply programming practices.	a, b, c	Various lab, assignments, all exams projects and quiz.
6. Comprehend and apply basic sorting and searching methods.	a, b, c	Various lab, assignments, all exams, projects and quiz.

Grading Policy: The final grade for this course will be based on attendance, labs, project, assignments, quizzes and exams, as described below:

- **Lecture Attendance: 10%.**
 - Each 1 point penalty will be given to each missing after the first two absences.
- **Labs: 30%.**
 - Lab attendance percentage: **10%**. Each 1 point penalty will be given to each missing after the first two absences.
 - Lab assignments percentage: **90%**.
 - Lab assignments' solutions will be provided by Lab TA after due.
- **Lab Project: 10%.**
 - One individual project.
 - Project's solution will be provided by Lab TA after due.
- **Assignments: 10%.**
 - Assignments' solutions will be provided by instructor after due.
- **Quizzes: 10%.**
 - The dates for quizzes will be announced before. There is no make-up for missing quiz.
 - The quiz starts at the very beginning of each lecture, and lasts for about 10 mins.
 - Quizzes' solutions will be provided by instructor after quiz.
- **Exams: 30%. (first exam: 10%; second exam: 10%; third exam: 10%)**
 - Students are required to **take exam on exam date**. There is no make-up for missing exam.
 - Exams' solutions will be provided by instructor after exam.

- Submissions grading policy:
 - All **lab assignments** and **project** MUST be submitted to **Blackboard**. **No Late submission allowed.**
 - All **writing assignments** MUST be submitted to **instructor in class**. **No Late submission allowed.**
 - Compile error – Receive **20%** of total assigned points.
 - Plagiarism – **0 point**. All the work should be finished **Individually**.
First time reported to instructor. Second time reported to department.
 - All code (functions, programs, etc.) must be submitted as instructed by TA or instructor.
 - Additional grading policy will be applied by TA or instructor.
 - The instructor or TA reserves the right to explain the confusing issues.
 - Programming environment: **Visual Studio 2015**
- The usual grading scale will be used:

A+	100 – 97
A	96.9 – 93
A-	92.9 – 90
B+	89.9 – 87
B	86.9 – 83
B-	82.9 – 80
C+	79.9 – 77
C	76.9 – 73
C-	72.9 – 70
D+	69.9 – 67
D	66.9 – 63
D-	62.9 – 60
F	Below 60

Ethical Conduct:

Although students are encouraged to discuss ideas and problems with the TA, instructor and other students, academic dishonesty will not be tolerated. **It is your responsibility to educate yourself about actions that constitute academic dishonesty.** If you are not sure whether a specific action is allowed, contact the instructor and/or the TA before you indulge in it! All submitted homework will be randomly checked for plagiarism. Academic dishonesty of any kind, if discovered, will result in a grade of 0 for the corresponding assignment. Any student who is caught indulging in academic dishonesty more than once will lead to a grade of “F” in the course, and further action according to the TTU operating procedures: <http://www.depts.ttu.edu/opmanual/OP34.12.pdf>

Classroom Civility:

All violations of classroom civility will be reported to the Student Judicial Programs. The Texas Tech University Catalog states: “Students are expected to assist in maintaining a classroom environment that is conducive to learning.” In order to ensure that all students gain from time spent in class, **students are prohibited from engaging in any form of distraction**, e.g., reading newspapers (or other articles), working on other courses, and using cell-phones for calls or messages. If you indulge in any such inappropriate behavior (without explicit consent of the instructor), you will (at the very least) be asked to leave the classroom.

Student with Disabilities:

Any student who, because of a disability, may require special arrangements in order to meet course requirements should contact the instructor as soon as possible to make any necessary arrangements. Students should present appropriate verification from Student Disability Services during the instructor's office hours. Please note that instructors are not allowed to provide classroom accommodations to a student until appropriate verification from Student Disability Services has been provided. For additional information, you may contact the Student Disability Services Office in 335 West Hall or 806-742-2405.

Course Schedule: This schedule is tentative and subject to change. All changes will be announced in class or on the course website (Blackboard).

• Aug 29:	Welcome
• Aug 31:	Chap01 – Introducing C
• Sep 02:	Chap02 – C Fundamentals
• Sep 05:	Chap02 – C Fundamentals
• Sep 07:	Chap03 – Formatted input/output; Lab begins
• Sep 09:	Chap03 – Formatted input/output
• Sep 12:	Quiz_1 (Ch01 – Ch03); Chap04 – Expressions
• Sep 14:	Chap04 – Expressions
• Sep 16:	Chap05 – Selection statements
• Sep 19:	Chap05 – Selection statements
• Sep 21:	Chap06 – Loops
• Sep 23:	Chap06 – Loops
• Sep 26:	Quiz_2 (Ch04 – Ch06); Chap07 – Basic types
• Sep 28:	Chap07 – Basic types
• Sep 30:	Chap08 – Array
• Oct 03:	First exam (Ch01 – Ch07)
• Oct 05:	Chap08 – Array
• Oct 07:	Chap09 – Functions
• Oct 10:	Chap09 – Functions
• Oct 12:	Chap10 – Program organization
• Oct 14:	Chap10 – Program organization
• Oct 17:	Quiz_3 (Ch08 – Ch10); Chap11 – Pointer
• Oct 19:	Chap11 – Pointer
• Oct 21:	Chap11 – Pointer
• Oct 24:	Chap12 – Pointer and array
• Oct 26:	Chap12 – Pointer and array
• Oct 28:	Chap12 – Pointer and array; Chap13 – String
• Oct 31:	Quiz_4 (Ch11 – Ch12); Chap13 – String
• Nov 02:	Chap14 – Preprocessor
• Nov 04:	Chap15 – Writing large program
• Nov 07:	Second exam (Ch08 – Ch15)
• Nov 09:	Chap16 – Structures, unions, and enumerations
• Nov 11:	Chap16 – Structures, unions, and enumerations
• Nov 14:	Chap17 – Advanced users of pointer
• Nov 16:	Chap18 – Declarations
• Nov 18:	Chap19 – Abstract data type
• Nov 21:	Quiz_5 (Ch16 – Ch18); Chap19 – Abstract data type
• Nov 23:	Thanksgiving holiday
• Nov 25:	Thanksgiving holiday
• Nov 28:	Chap22 – Input/output
• Nov 30:	Chap22 – Input/output; Last lab
• Dec 02:	Unix programming
• Dec 05:	Unix programming
• Dec 07:	Last day of classes
• Dec 09:	No class
• Dec 12:	Third exam (Ch16 – Ch22) 7:30AM – 10:00 AM