ET156
Introduction to C Programming
[Onsite]

Course Description:

This course is designed to help students with the fundamental concepts and terminology of computer programming and practical skills in designing, writing and debugging simple computer programs in C.

Prerequisite(s) and/or Corequisite(s):

Prerequisites: TB143 Introduction to Personal Computers or equivalent

Credit hours: 4

Contact hours: 50 (30 Theory Hours, 20 Lab Hours)

Syllabus: Introduction to C Programming

Instructor: _____
Office hours: _____
Class hours: _____

Major Instructional Areas

1. Program structure and keywords
2. Variables and data types
3. Error handling and debugging
4. Conditional structures
5. Loops and recursion
6. Functions
7. Arrays
8. Strings
9. File processing

Course Objectives

1. Create a C program that performs input, processing, and output.
2. Write a program that uses variables and constants.
3. Use debugging techniques to locate and correct programming errors.
4. Create a program that uses conditional statements to solve a problem.
5. Create a program that uses loops to solve a problem.
6. Apply modular programming techniques to C programming.
7. Use arrays to store and manipulate data.

8. Write a program that processes characters and strings.
9. Write a program that reads and writes data to a file.
10. Apply C programming to solve an electronics system or process.

SCANS Objectives

SCANS is an acronym for Secretary's Commission on Achieving Necessary Skills. The committee, created by the National Secretary of Labor in the early 1990s, created a list of skills and competencies that the committee feels are necessary for employees to function in a high-tech job market.

1. Acquire information.
2. Understand how technological systems work and operate effectively.
3. Demonstrate competence in understanding systems.
4. Know how a system's structures relate to goals.
5. Demonstrate competence in selecting technology and determining desired outcomes and applicable constraints.
6. Demonstrate competence in how to apply technology to tasks.

Course Outline

Note: All graded activities, except the Project and Final Exam, are listed below in the pattern of <Unit Number>.<Assignment Number>. For example, Lab 5.1 refers to the 1st lab activity in Unit 5.

Unit	Activities
1– Introduction to C Programming: The C Compiler, Variables, Memory, Input, and Output	• Content Covered: *Problem Solving and Program Design in C:* o Chapter 1, "Overview of Computers and Programming" o Chapter 2, "Overview of C," pp. 46-70, Sections 2.1-2.4 • Assignments: 1.1 or 1.2 • Labs: 1.1
2– Arithmetic	• Read from *Problem Solving and Program Design in C:* o Chapter 2, "Overview of C," pp. 70-100, Section 2.5

Expressions and Library Functions	through Chapter Review o Chapter 3, “Top-Down Design with Functions,” pp. 105-126, Sections 3.1-3.3 • Assignments: 2.1 or 2.2 • Labs: 2.1 • Project (Assigned)
3– Selection Structures: single & dual choices	• Read from *Problem Solving and Program Design in C:* o Chapter 4, “Selection Structures: if and switch Statements,” pp. 155-188, Sections 4.1-4.5 • Assignments: 3.1 or 3.2 • Labs: 3.1, 3.2 • Quizzes: 3.1 • Project Part 1: Due

4– Selection Structures: Nested if & switch Statements	• Read from *Problem Solving and Program Design in C:* ○ Chapter 4, “Selection Structures: else-if and switch Statements,” pp.191-209, Section 4.7 through Chapter Review • Assignments: 4.1 • Labs: 4.1, 4.2 • Exams: 4.1 • Project Part 2: Due
5– Repetition Structures: Determinate or Counting loops	• Read from *Problem Solving and Program Design in C:* ○ Chapter 5, “Repetition and Loop Statements,” pp. 221-243, Sections 5.1-5.4 • Assignments: 5.1, 5.2 • Labs: 5.1 • Project Part 3: Due
6– Repetition Structures : Indeterminate or Conditional loops	• Read from *Problem Solving and Program Design in C:* ○ Chapter 5, “Repetition and Loop Statements,” pp. 243-281, Section 5.5 through the Chapter Review • Assignments: 6.1, 6.2 • Labs: 6.1 • Quizzes: 6.1 • Project Part 4: Due

7–
 	
Functions and
 	
Modular Programming
	• Read from *Problem Solving and Program Design in C:*
	  ○ Chapter 3, “Top-Down Design with Functions,” pp. 126-150, Section 3.4 through Chapter Review
	  ○ Chapter 6, “Modular Programming”
	• Assignments: 7.1, 7.2
	• Labs: 7.1

	• Exams: 7.1 • Project Part 5: Due
8– Data Types and Arrays	• Read from *Problem Solving and Program Design in C:* o Chapter 7, “Simple Data Types” o Chapter 8, “Arrays” • Assignments: 8.1, 8.2 • Labs: 8.1, 8.2, 8.3, 8.4, 8.5 • Project Part 6: Due
9– Strings and Tracing Recursions	• Read from *Problem Solving and Program Design in C:* o Chapter 9, “Strings” o Chapter 10, “Recursion,” pp. 515-529, Sections 10.1-10.2 • Assignments: 9.1, 9.2 • Labs: 9.1, 9.2 • Project Part 7: Due
10– File Processing	• Read from *Problem Solving and Program Design in C:* o Chapter 12, “Text and Binary File Processing” • Assignments: 10.1 • Labs: 10.1 • Quizzes: 10.1 • Project Part 8: Due
11– Review and Final Exam	• Final Exam • Project Part 9: Due

Instructional Methods

The course uses a variety of instructional methods, including lectures, in-class activities, hands-on programming practice, and assignments to teach programming logic and fundamental C programming techniques.

You will learn to analyze a problem and identify the steps needed to solve the problem, and you will write pseudocode and algorithms to describe the steps. You will also learn how to use flowcharts to illustrate the steps graphically.

Hands-on labs will give you practice writing C code that instructs the computer to receive input, process data, and output results. The focus of the course will be on using C to solve programming problems.

A project will allow you to demonstrate the skills you learn by designing and building a small program.

Exams and quizzes will evaluate your understanding of the core concepts covered in this course, culminating in a final exam at the end of the course.

Instructional Materials and References

Student Textbook Package

- Hanly, Jeri R., and Elliot B. Koffman. *Problem Solving and Program Design in C. 6th ed. Boston: Addison-Wesley, 2010*
- Introduction to C Programming Student CD

  (This CD contains Pelles C)

Other Required Resources

In addition to the student textbook package, the following equipment and tools are also required in this course:

- Microsoft Office

- Microsoft Visio
- Student source code (available for download in the ITT Tech Virtual Library: School of Electronics Technology> Recommended Links> ET156 course materials)
- Answers to odd questions (available for download in the ITT Tech Virtual Library: School of Electronics Technology> Recommended Links> ET156 course materials)
- Walls, Colin. *Embedded Software: The Works. Burlington, MA: Newnes, 2006. (Chapter 1)* ITT Tech Virtual Library> Books> 24x7.

References

ITT Tech Virtual Library

Log on to the ITT Tech Virtual Library at http://library.itt-tech.edu/ to access online books, journals, and other reference resources selected to support ITT Tech curricula.

Books

You may click "Books" or use the "Search" function on the home page to find the following books.

- Books 24x7
  - Biafore, Bonnie. *Visio 2003 Bible. Indianapolis: John Wiley & Sons, 2004.*
  - Franek, Frantisek. *Memory as a Programming Concept in C and C++. New York: Cambridge University Press, 2004.*
  - Gookin, Dan. *C for Dummies. 2nd ed. Hoboken, NJ: John Wiley & Sons, 2004.*
  - Hodges, M. Susan. *Computers: Systems, Terms and Acronyms. 17th ed. Casselberry, FL: SemCo Enterprises Inc., 2007.*
  - Horton, Ivor. *Beginning C: From Novice to Professional. 4th ed. Berkeley, CA: Apress, 2006.*
  - Lemke, Judy. *Microsoft Office Visio 2003 Step by Step. Redmond, WA: Microsoft Press, 2005.*
  - Schildt, Herbert. *C: The Complete Reference. 4th ed. Berkeley, CA: McGraw Hill/Osborne, 2000.*

- Vine, Michael. *C Programming for the Absolute Beginner: The Fun Way to Learn Programming. Indianapolis: Premier Press, 2002.*
- Walls, Colin. *Embedded Software: The Works. Burlington, MA: Newnes, 2006.*

- NetLibrary

- Kochan, Stephen G. *Programming in C. 3rd ed. Indianapolis: Pearson Education Inc., 2005.*
- Prata, Stephen. *C Primer Plus. 5th ed. Indianapolis: Pearson Education Inc., 2005.*

Periodicals

You may click “Periodicals” or use the “Search” function on the home page to find the following periodicals.

- ProQuest Science Journals> Dr. Dobbs

Click School of Information Technology on the home page.

- Professional Organizations
  - Association of C & C++ Users
  - Business Software Alliance
- Recommended Links
  - Free Programming Resources
  - Tech Fest
- Tutorial Links
  - Computer Science Tutorials
  - Computer Technical Tutorials
  - Edumax
  - Programming Tutorials
  - Tutorialized

Course Evaluation and Grading

Evaluation Criteria Table

The final grades will be based on the following categories:

CATEGORY	WEIGHT
Exams	20%
Quizzes	15%
Labs	20%
Assignments	10%
Project	15%
Final Exam	20%
Total	**100%**

Note: Students are responsible for abiding by the Plagiarism Policy.

Grade Conversion Table

The final grades will be calculated from the percentages earned in the course, as follows:

A	90-100%	4.0
B+	85-89%	3.5
B	80-84%	3.0
C+	75-79%	2.5
C	70-74%	2.0
D+	65-69%	1.5
D	60-64%	1.0

F	<60%	0.0

(End of Syllabus)