ITT Technical Institute

ET2560

Introduction to C Programming

Onsite and Online Course

SYLLABUS

Credit hours: 4.5

Contact/Instructional hours: 56 (34 Theory Hours, 22 Lab Hours)

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

Prerequisites: NT1110 Computer Structure and Logic or equivalent

Course Description:

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

COURSE SUMMARY

COURSE DESCRIPTION

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

MAJOR INSTRUCTIONAL AREAS

1. Algorithms and flowcharting
2. Program structure and keywords
3. Variables and data types
4. Use of an integrated development environment (IDE)
5. Operators and expressions
6. Input validation and error handling
7. Conditional and looping structures
8. Functions
9. Debugging techniques
10. Arrays
11. Strings
12. File processing

COURSE LEARNING OBJECTIVES

By the end of this course, you should be able to:

1. Describe the importance of programmable computers in solving engineering problems and the role of programs in control systems.
2. Describe the methods of creation, maintenance, and modification of computer programs.
3. Use a flowchart to express an algorithm that uses a computer to solve a simple problem.
4. Design a C program algorithm that solves a problem.
5. Use an integrated development environment (IDE) to develop a C program.
6. Develop a functional C program that implements a design.

7. Apply the modular programming techniques of functions and local and global variables in a C program.
8. Use debugging techniques to correct programming errors.
9. Analyze the inputs, operation, and outputs of a C subroutine or program.
10. Apply C programming to solve a complex problem relevant to electronic circuits or control.

COURSE OUTLINE

MODULE 1: ALGORITHM, FLOWCHARTS, AND PSEUDOCODE

COURSE LEARNING OBJECTIVES COVERED

- Describe the importance of programmable computers in solving engineering problems and the role of programs in control systems.
- Describe the methods of creation, maintenance, and modification of computer programs.
- Use a flowchart to express an algorithm that uses a computer to solve a simple problem.
- Design a C program algorithm that solves a problem.

TOPICS COVERED

- Control Systems
- Computational Mathematics
- Iterative Method of Solution
- Software Development Model
- Compiler
- Programming Language Interpreter
- Algorithm
- Pseudocode
- Flowchart of Software Algorithm

MODULE LEARNING ACTIVITIES	GRADED	OUT-OF-CLASS TIME
Reading: Hanly & Koffman, Chapter 1 (pp. 14–39).	No	3 hr
Lesson: Study the lesson for this module.	No	2 hr
Discussion: Participate in the discussion titled "Overview of Programming."	Yes	1 hr
Lab: Complete the lab titled "Flowcharts."	Yes	N/A
Project: Read and begin the project.	No	1 hr

Total Out-Of-Class Activities: 7 Hours

MODULE 2: PROGRAMMING CALCULATIONS AND STANDARDIZATION

COURSE LEARNING OBJECTIVES COVERED

- Use an integrated development environment (IDE) to develop a C program.
- Develop a functional C program that implements a design.
- Apply the modular programming techniques of functions and local and global variables in a C program.
- Use debugging techniques to correct programming errors.

TOPICS COVERED

- C Programming Language
- Embedded C
- Library Functions–*printf and scanf*
- Data Types–Integer and Floating Point
- Types of Errors
- Arrays

MODULE LEARNING ACTIVITIES	GRADED	OUT-OF-CLASS TIME
Reading: Hanly & Koffman, Chapter 2 (pp. 46–99), Chapter 3 (pp. 107–124), and Chapter 7 (pp. 406–414).	No	9 hr
Lesson: Study the lesson for this module.	No	2.5 hr
Discussion: Participate in the discussion titled "Operators and Statements."	Yes	1 hr
Exercise: Submit the exercise titled "Overview of C and Programming Calculations."	Yes	2.5 hr
Lab 1: Complete the lab titled "Algorithm and Flowcharts."	Yes	N/A
Lab 2: Complete the lab titled "Working with the Pelles C IDE and Programming Calculations."	Yes	N/A
Project: Continue work on Project Part 1.	No	3 hr

Total Out-Of-Class Activities: 18 Hours

MODULE 3: DECISION STRUCTURES AND PROGRAM CODES

COURSE LEARNING OBJECTIVES COVERED

- Describe the methods of creation, maintenance, and modification of computer programs.
- Develop a functional C program that implements a design.
- Use debugging techniques to correct programming errors.
- Analyze the inputs, operation, and outputs of a C subroutine or program.

TOPICS COVERED

- Top-Down Design
- Modular Programming
- *if Statement*
- *While Loop*
- *Do-While Loop*
- Break Statement
- Continue Statement

MODULE LEARNING ACTIVITIES	GRADED	OUT-OF-CLASS TIME
Reading: Hanly & Koffman, Chapter 3 (pp. 124–163) and Chapter 5 (pp. 235–298).	No	11 hr
Lesson: Study the lesson for this module.	No	2.5 hr
Exercise 1: Submit the exercise titled "Programming Decisions and Loops."	Yes	2 hr
Exercise 2: Submit the exercise titled "Repetition and Loop Statement."	Yes	2 hr
Lab: Complete the lab titled "Working with *if* Condition Statements and Loop Statements."	Yes	N/A
Project: Submit Project Part 1.	Yes	3 hr

Total Out-Of-Class Activities: 20.5 Hours

MODULE 4: FUNCTIONS, ARRAYS, AND STRINGS

COURSE LEARNING OBJECTIVES COVERED

- Design a C program algorithm that solves a problem.
- Develop a functional C program that implements a design.
- Apply the modular programming techniques of functions and local and global variables in a C program.
- Use debugging techniques to correct programming errors.

TOPICS COVERED

- Modular Programming
- Function Arguments
- Arrays in C Language
- Array Searching and Sorting Algorithms
- Strings in C Language

MODULE LEARNING ACTIVITIES	GRADED	OUT-OF-CLASS TIME
Reading: Hanly & Koffman, Chapter 6 (pp. 315–359), Chapter 7 (pp. 375–405 and pp. 414–438) and Chapter 8 (pp. 453–460).	No	12 hr
Lesson: Study the lesson for this module.	No	2.5 hr
Discussion: Participate in the discussion titled "Arrays."	Yes	1 hr
Exercise: Submit the exercise titled "Functions with Output Parameters."	Yes	2 hr
Lab 1: Complete the lab titled "Working with Program Variable and Functions."	Yes	N/A
Lab 2: Complete the lab titled "Debugging a Program."	Yes	N/A
Project: Continue work on Project Part 2.	No	3 hr

Total Out-Of-Class Activities: 20.5 Hours

MODULE 5: STRINGS, FILES, AND RECURSIVE FUNCTIONS

COURSE LEARNING OBJECTIVES COVERED

- Use an integrated development environment (IDE) to develop a C program.
- Develop a functional C program that implements a design.
- Apply the modular programming techniques of functions and local and global variables in a C program.
- Use debugging techniques to correct programming errors.
- Analyze the inputs, operation, and outputs of a C subroutine or program.

TOPICS COVERED

- Recursive Functions
- *fscanf and fprintf Statements*
- C Language File I/O
- Binary and Text Files
- Database

MODULE LEARNING ACTIVITIES	GRADED	OUT-OF-CLASS TIME
Reading: Hanly & Koffman, Chapter 9 (pp. 518–537) and Chapter 11 (pp. 623–651).	No	5.5 hr
Lesson: Study the lesson for this module.	No	2.5 hr
Exercise 1: Submit the exercise titled “Strings.”	Yes	2.5 hr
Exercise 2: Submit the exercise titled “Recursive Functions.”	Yes	2.5 hr
Lab 1: Complete the lab titled “Working with Strings and Recursive Functions.”	Yes	N/A
Lab 2: Complete the lab titled “Programming Functions and Using File Processing.”	Yes	N/A
Project: Continue work on Project Part 2.	No	3 hr
Final Exam: Prepare for the final exam.	No	5 hr

Total Out-Of-Class Activities: 21 Hours

MODULE 6: COURSE REVIEW AND FINAL EXAM

COURSE LEARNING OBJECTIVES COVERED

- Describe the importance of programmable computers in solving engineering problems and the role of programs in control systems.
- Describe the methods of creation, maintenance, and modification of computer programs.
- Use a flowchart to express an algorithm that uses a computer to solve a simple problem.
- Design a C program algorithm that solves a problem.
- Use an integrated development environment (IDE) to develop a C program.
- Develop a functional C program that implements a design.
- Apply the modular programming techniques of functions and local and global variables in a C program.
- Use debugging techniques to correct programming errors.
- Analyze the inputs, operation, and outputs of a C subroutine or program.
- Apply C programming to solve a complex problem relevant to electronic circuits or control.

TOPICS COVERED

- Identifiers
- Boolean Expressions
- Control Structures
- Input/Output

MODULE LEARNING ACTIVITIES	GRADED	OUT-OF-CLASS TIME
Reading: ITT Tech Virtual Library > School of Study > School of Information Technology > Tutorial Links > Computer Science Tutorials > Programming Languages > Programs.	No	3 hr
Lesson: Study the lesson for this module.	No	1 hr
Project: Submit Project Part 2.	Yes	3 hr
Final Exam: Take the final exam.	Yes	N/A

Total Out-Of-Class Activities: 7 Hours

EVALUATION AND GRADING

EVALUATION CRITERIA

The graded assignments will be evaluated using the following weighted categories:

CATEGORY	WEIGHT
Discussion	10%
Exercise	20%
Lab	30%
Project	20%
Final Exam	20%
TOTAL	100%

GRADE CONVERSION

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

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

)	
F (0.0)	<60%

LEARNING MATERIALS AND REFERENCES

REQUIRED RESOURCES

COMPLETE TEXTBOOK PACKAGE

- Hanly, J. R., & Koffman, E. B. (2013). *Problem solving and program design in C (7th ed.)*. Boston, MA: Addison-Wesley.

OTHER ITEMS

- Microsoft Visio 2007 (or later)
- Pelles C Complier for Microsoft Windows (http://www.smorgasbordet.com/pellesc/)

Note: You can download Microsoft Visio from the DreamSpark website. Refer to the DreamSpark Installation Guide for download instructions.

RECOMMENDED RESOURCES

- Books and Professional Journals
  - Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). *The design and analysis of computer algorithms. Reading, MA: Addison-Wesley.*
  - American National Standards Institute. (1990). *American national standard for information systems - Programming language - C: ANSI X3.159-1989. New York, NY: American National Standards Institute.*
  - Fogler, H. S., & LeBlanc, S. E. (2008). *Strategies for creative problem solving. Boston, MA: Pearson Custom Publishing.*
  - Gaddis, T. (2008). *Starting out with programming logic and design (Custom ed.). Boston, MA: Addison-Wesley.*
  - Hanly, J. R., & Koffman, E. B. (2001). *C program design for engineers (2nd ed.). Boston, MA: Addison-Wesley Longman.*
  - Kelley, A., & Pohl, I. (1998). *A book on C: Programming in C (4th ed.). Boston, MA: Pearson Custom Publishing.*
  - Kernighan, B. W., & Ritchie, D. M. (1988). *The C programming language (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.*
  - Plauger, P. J., & Brodie, J. (1989). *Standard C. Redmond, WA: Microsoft Press.*

- Ward, R. (1989). *A programmer's introduction to debugging C. Lawrence, KS: R & D Publications.*

- ITT Tech Virtual Library (accessed via Student Portal | https://studentportal.itt-tech.edu)
  - ITT Tech Virtual Library > School of Study > School of Information Technology > Tutorial Links > Computer Science Tutorials > Programming Languages >
    - Identifiers
    - Assignment
    - Expressions
    - Boolean Expressions
    - Selection
    - Loops
    - Subprograms
    - Parameters
- Other References
  - C Programming
    http://www.cprogramming.com (accessed on 08/11/14)
    The website includes various links for C programming language.
  - RFFlow Professional Flowcharting
    http://www.rff.com (accessed on 08/11/14)
    The website provides information on tools for drawing flowcharts and other types of diagrams.
  - C Tutorial
    http://fresh2refresh.com/c-tutorial-for-beginners/ (accessed on 08/11/14)
    The website provides tutorial lessons for C programming.

INSTRUCTIONAL METHODS AND TEACHING STRATEGIES

The curriculum employs a variety of instructional methods that support the course objectives while fostering higher cognitive skills. These methods are designed to encourage and engage you in the learning process in order to maximize learning opportunities. The instructional methods include but are not limited to lectures, collaborative learning options, use of technology, and hands-on activities.

To implement the above-mentioned instructional methods, this course uses several teaching strategies, such as hands-on labs, exercises, and discussions. Your progress will be regularly assessed through a variety of assessment tools including discussions, exercises, labs, project, and the final exam.

OUT-OF-CLASS WORK

For purposes of defining an academic credit hour for Title IV funding purposes, ITT Technical Institute considers a quarter credit hour to be the equivalent of: (a) at least 10 clock hours of classroom activities and at least 20 clock hours of outside preparation; (b) at least 20 clock hours of laboratory activities; or (c) at least 30 clock hours of externship, practicum or clinical activities. ITT Technical Institute utilizes a “time-based option” for establishing out-of-class activities which would equate to two hours of out-of-class activities for every one hour of classroom time. The procedure for determining credit hours for Title IV funding purposes is to divide the total number of classroom, laboratory, externship, practicum and clinical hours by the conversion ratios specified above. A clock hour is 50 minutes.

A credit hour is an artificial measurement of the amount of learning that can occur in a program course based on a specified amount of time spent on class activities and student preparation during the program course. In conformity with commonly accepted practice in higher education, ITT Technical Institute has institutionally established and determined that credit hours awarded for coursework in this program course (including out-of-class assignments and learning activities described in the “Course Outline” section of this syllabus) are in accordance with the time-based option for awarding academic credit described in the immediately preceding paragraph.

ACADEMIC INTEGRITY

All students must comply with the policies that regulate all forms of academic dishonesty or academic misconduct. For more information on the academic honesty policies, refer to the Student Handbook and the School Catalog.

INSTRUCTOR DETAILS

Instructor Name	
Office Hours	
Contact Details	

(End of Syllabus)