ITT Technical Institute

ET4560

C++ Programming

Onsite Course

SYLLABUS

Credit hours: 4.5

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

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

Prerequisites: ET2560 Introduction to C Programming or equivalent

Course Description:

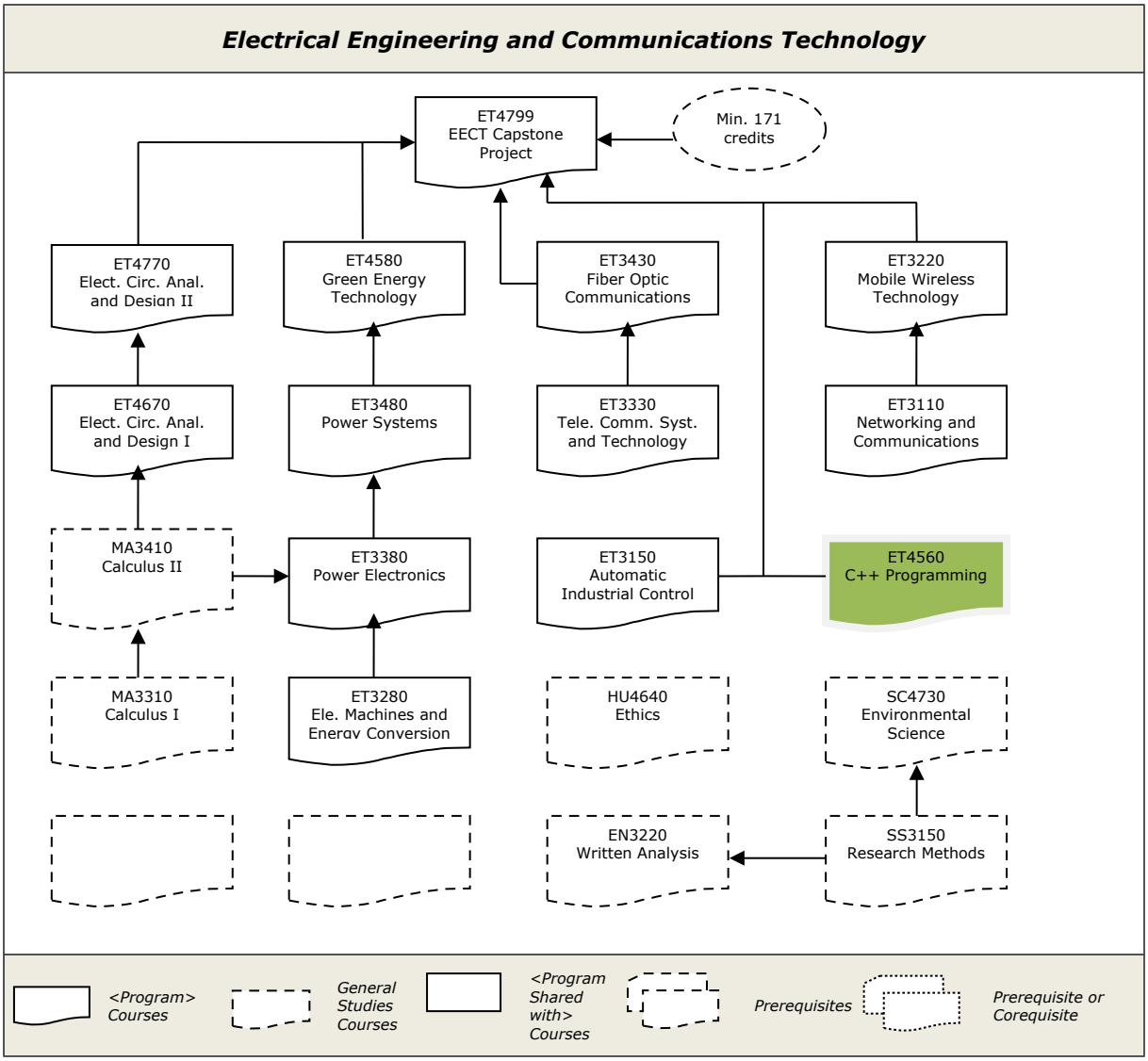
This course introduces concepts of object oriented programming and provides hands-on exercises in C++ programming. Areas of instruction include primitive data types, control structures, functions, pass-by-value, pass-by-reference, array, pointers, C-strings, recursion, class and objects, file input and output, operator overloading and inheritance.

Where Does This Course Belong?

This course is required for the Electrical Engineering and Communications Technology program. This program covers the following core areas:

- Process control
- Embedded systems
- Electronic circuit analysis and design
- Data and network communications
- Telecommunications and mobile wireless technology
- Fiber optic communications
- Electrical machines and energy conversion
- Power electronics and power systems
- Green energy technology
- Computer programming

The following diagram demonstrates how this course fits in the program:



NOTE: *Refer to the catalog for the state-specific course information, if applicable.*

Course Summary

Major Instructional Areas

1. C++ program structure
2. Data types, operators, and expressions
3. Decisions, loops, and, recursion
4. Functions
5. Classes, objects, and unions
6. Arrays and strings
7. Searching and sorting
8. Pointers
9. Inheritance and polymorphism
10. File input and output
11. Exceptions
12. Standard Template Library (STL)

Course Objectives

1. Solve mathematical problems by coding a program which uses variables, strings, operators, and expressions.
2. Demonstrate the use of various data types.
3. Solve mathematical and Boolean problems by coding the various control structures within a program.
4. Demonstrate modular programming techniques.
5. Demonstrate object-oriented programming techniques.
6. Process data by using arrays.
7. Process text data.
8. Perform file input and output.
9. Demonstrate debugging and exception-handling techniques.
10. Demonstrate the use of pointers.
11. Demonstrate use of the Standard Template Library (STL).
12. Code a search function and a sort function.
13. Demonstrate the use of inheritance, recursion, and polymorphism.

Learning Materials and References

Required Resources

Textbook Package	New to this Course	Carried over from Previous Course(s)	Required for Subsequent Course(s)
Gaddis, T., Walters, J., & Muganda, G. (2014). *Starting Out with C++ Early Objects* (8th ed. with CD). Boston, MA: Addison-Wesley.			
Other Items	**New to this Course**	**Carried over from Previous Course(s)**	**Required for Subsequent Course(s)**
Removable hard drive with virtual image named ITT-LAB that contains Visual Studio 2005			

Recommended Resources

ITT Tech Virtual Library (accessed via Student Portal)

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.

Basic Search

- Davis, S. R. (2009). *C++ for dummies*. (6th ed.). Hoboken, NJ: John Wiley & Sons.
- Duffy, D. J. (2006). *Introduction to C++ for financial engineers: An object-oriented approach*. Hoboken, NJ: John Wiley & Sons.
- Horton, I. (2008). *Ivor Horton's beginning Visual C++ 2008.* Hoboken, NJ: Wrox Press.
- Hughes, C., & Hughes, T. (2008). *Professional multicore programming: Design and implementation for C++ developers.* Hoboken, NJ: Wrox Press.
- Lee, M. (2009). *C++ programming for the absolute beginner*. (2nd ed.). Boston, MA: Cengage Learning.
- Lischner, R. (2009). *Exploring C++: The programmer's introduction to C++*. New York, NY: Apress.
- Miller, A., & Ford, Jr., J. L. (2006). *Microsoft Visual C++ 2005 Express edition programming for the absolute beginner*. Boston, MA: Cengage Course Technology.
- Schildt, H. (2008). *Herb Schildt's C++ programming cookbook*. New York, NY: McGraw Hill/Osborne.

Other References

The following resource may be found **outside** of the ITT Tech Virtual Library.

Web site

- Microsoft Developer Network
  http://msdn.microsoft.com
  This site provides videos, virtual labs, tutorials, and reference information for beginners, plus developer tools for the Microsoft family of products.

NOTE: All links are subject to change without prior notice.

Information Search

Use the following keywords to search for additional online resources that may be used for supporting your work on the course assignments:

- Pseudocode
- Algorithm
- ASCII code
- Encryption
- Creating an array
- Function library
- Binary search
- Boolean variable
- High-level language
- Compiling multi-file programs
- Recursion
- Syntax
- Logic
- For loop
- While/Do-While loop
- Header file
- Forward iterator
- Initialization expression

- If/then/else
- JavaScript
- Machine language
- Control statement
- Fibonacci
- Object-oriented programming

Course Plan

Suggested Learning Approach

In this course, you will be studying individually and within a group of your peers. As you work on the course deliverables, you are encouraged to share ideas with your peers and instructor, work collaboratively on projects and team assignments, raise critical questions, and provide constructive feedback.

Use the following advice to receive maximum learning benefits from your participation in this course:

DO	DON'T
▪ Do take a proactive learning approach. ▪ Do share your thoughts on critical issues and potential problem solutions. ▪ Do plan your course work in advance. ▪ Do explore a variety of learning resources in addition to the textbook. ▪ Do offer relevant examples from your experience. ▪ Do make an effort to understand different points of view. ▪ Do connect concepts explored in this course to real-life professional situations and your own experiences.	▪ Don't assume there is only one correct answer to a question. ▪ Don't be afraid to share your perspective on the issues analyzed in the course. ▪ Don't be negative about the points of view that are different from yours. ▪ Don't underestimate the impact of collaboration on your learning. ▪ Don't limit your course experience to reading the textbook. ▪ Don't postpone your work on the course deliverables – work on small assignment components every day.

Course Outline

Unit 1: INTRODUCTION TO C++

Out-of-class work: 9 hours

Upon completion of this unit, students are expected to:

- Examine the development environment, debugger, and other tools for C++ programming.
- Declare and use variables and data types.
- Write code using various operators.
- Design a C++ program.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 1, Sections 1.4-1.7, pp. 13-23; Chapter 2, Sections 2.1-2.15, pp. 27-64	Lab	Unit 1 Lab 1: Design and Implement	2.27%
		Unit 1 Lab 2: Programming Challenges	2.27%
	Assignment	Unit 1 Assignment 1: Review Questions	1.5%

Unit 2: EXPRESSIONS AND FILE I/O

Out-of-class work: 9 hours

Upon completion of this unit, students are expected to:

- Code solutions to mathematical problems.
- Code solutions to Boolean problems.
- Input data from file to program and output solution from program to file.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 3, Sections 3.1-3.12, pp. 75-136; Chapter 4, Section 4.1, pp.155-160	Assignment	Unit 2 Assignment 1: Review Questions	1.5%
	Lab	Unit 2 Lab 1: Programming Challenges	2.28%

Unit 3: DECISIONS, LOOPS, AND RECURSIONS

Out-of-class work: 9 hours

Upon completion of this unit, students are expected to:

- Code decisions and comparisons using various program structures.
- Code loops of varying program structures to solve problems with repetitive algorithms.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)

• Gaddis, Walters, and Muganda, Chapter 4, Sections 4.2-4.14, pp. 160-224; Chapter 5, Sections 5.1-5.13, pp. 243-304; Chapter 14, Sections 14.1-14.2, pp. 899-908	Assignment	Unit 3 Assignment 1: Review Questions	1.5%
	Lab	Unit 3 Lab 1: Programming Challenges	2.27%

Unit 4: MODULAR PROGRAMMING

Out-of-class work: 12 hours

Upon completion of this unit, students are expected to:

- Design and use functions.
- Demonstrate pass-by-value and pass-by-reference.
- Demonstrate recursion.
- Declare and use local and global variables.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 6, Sections 6.1-6.13, pp. 323-377; Chapter 14, Sections 14.3-14.4, pp. 908-910	Assignment	Unit 4 Assignment 1: Review Questions	1.5%
	Lab	Unit 4 Lab 1: Programming Challenges	2.28%
	Project	Project 1	10%

Unit 5: INTRODUCTION TO OBJECTS AND CLASSES

Out-of-class work: 9 hours

Upon completion of this unit, students are expected to:

- Design and use objects.
- Declare and use classes.

READING ASSIGNMENT	GRADED ACTIVITIES/DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 7, Sections 7.1-7.7, pp. 407-432	Assignment	Unit 5 Assignment 1: Review Questions	1.5%
	Lab	Unit 5 Lab 1: Programming Challenges	2.27%

Unit 6: OBJECT-ORIENTED PROGRAMMING

Out-of-class work: 12 hours

Upon completion of this unit, students are expected to:

- Pass objects to a function.

- Declare and use unions.

READING ASSIGNMENT	GRADED ACTIVITIES/DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 7, Sections 7.8-7.13, pp. 432-473	Assignment	Unit 6 Assignment 1: Review Questions	1.5%
	Lab	Unit 6 Lab 1: Programming Challenges	2.28%
	Exam	Midterm Exam	10%

Unit 7: ARRAYS AND STL

Out-of-class work: 12 hours

Upon completion of this unit, students are expected to:

- Declare and demonstrate an array.
- Input data to an array.
- Output array contents.
- Process array contents.
- Define and use the vector data type using STL.

READING ASSIGNMENT	GRADED ACTIVITIES/DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 8, Sections 8.1-8.9, pp. 503-552; Chapter 16, Section 16.5, pp. 1000-1013	Assignment	Unit 7 Assignment 1: Review Questions	1.5%
	Lab	Unit 7 Lab 1: Programming Challenges	2.27%
	Project	Project 2	10%

Unit 8: SEARCH AND SORT

Out-of-class work: 9 hours

Upon completion of this unit, students are expected to:

- Search an array.
- Sort an array.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 9, Sections 9.1-9.5, pp. 595-619	Assignment	Unit 8 Assignment 1: Review Questions	1.5%
	Lab	Unit 8 Lab 1: Programming Challenges	2.27%

Unit 9: POINTERS

Out-of-class work: 9 hours

Upon completion of this unit, students are expected to:

- Process an array using pointer notation.

- Demonstrate proper exception handling.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 10, Sections 10.1-10.6, pp. 637-652; Chapter 16, Section 16.1, pp. 971-983	Assignment	Unit 9 Assignment 1: Review Questions	1.5%
	Lab	Unit 9 Lab 1: Programming Challenges	2.27%

Unit 10: INHERITANCE AND POLYMORPHISM

Out-of-class work: 9 hours

Upon completion of this unit, students are expected to:

- Demonstrate and use inheritance.
- Demonstrate and use polymorphism.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
• Gaddis, Walters, and Muganda, Chapter 11, Section 11.10, pp. 758-763; Chapter 15, Sections 15.1-15.2, pp. 935-944	Assignment	Unit 10 Assignment 1: Review Questions	1.5%
	Lab	Unit 10 Lab 1: Programming Challenges	2.27%

Unit 11: REVIEW AND EXAM

Out-of-class work: 12 hours

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
None	Exam	Final Exam	15%
	Project	Final Project (PORTFOLIO)	15%

Note: Your instructor may add a few learning activities that will change the grade allocation for each assignment in a category. The overall category percentages will not change.

Evaluation and Grading

Evaluation Criteria

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

Category	Weight
Assignment	15%
Lab	25%
Project	35%
Exam	25%
TOTAL	**100%**

Grade Conversion

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

Grade	Percentage	Credit
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

Academic Integrity

All students must comply with the policies that regulate all forms of academic dishonesty, or academic misconduct, including plagiarism, self-plagiarism, fabrication, deception, cheating, and sabotage. For more information on the academic honesty policies, refer to the Student Handbook and the Course Catalog.

(End of Syllabus)