ITT Technical Institute

IT106T

Programming in C++ I

Onsite Course

SYLLABUS

Credit hours: 4

Contact/Instructional hours: 60 (36 Theory Hours, 24 Lab Hours)

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

Prerequisites: IT104T Introduction to Computer Programming

Course Description:

Students will write, enter, run and debug programs using the C++ language. Topics include simple C++ operations, functions, procedures and data operations.

SYLLABUS

Instructor: _____

Office hours: _____

Class hours: _____

Major Instructional Areas

1. Introduction to C++ Programming
2. Controlling Execution
3. Functions
4. Classes and Objects
5. Class Inheritance
6. Exception Handling
7. Arrays
8. Pointers and Strings
9. Streams Processing

Course Objectives

1. Identify the various features of C++.
2. Use the C++ development environment tools.
3. Identify the basic structure of a C++ program.
4. Perform input and output operations.
5. Declare and use variables of various data types.
6. Use operators to manipulate data.
7. Use conditional control statements.
8. Use control statements to implement repetition.
9. Create and utilize user-defined functions.
10. Create and use pointers.
11. Use single and multi-dimensional arrays.
12. Declare and manipulate strings.
13. Create classes that expose member functions.
14. Write code to manage object creation and destruction.
15. Create a base class and a derived class.
16. Implement exception handling.

Teaching Strategies

In this course, the students will be given opportunities to design, write, and test C++ console applications. This will help them use the structure and syntax for creating a C++ program. This course introduces Unified Modeling Language (UML) to help them decide how to design an object-oriented program.

The following are some of the strategies used in the course:
- Group activities in which students analyze a specific issue or a C++ technique and present their findings to the class.
- Code demonstrations.
- Open-ended questions to initiate discussions in class.
- Labs and projects that allow students significant hands-on practice.

- Each unit includes at least one lab exercise that reinforces the content specific to that unit. The nature of programming requirements is such that in the later units, students will need to apply concepts presented earlier in the course.
- Each unit includes a homework assignment in which students answer questions about on topics covered and apply their knowledge to programming and debugging problems.

Course Resources

Student Textbook Package
- (Harvey & Paul) Deitel & Associates. *C++ How to Program*. 6th Ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2008.

References and Resources

ITT Tech Virtual Library
Login to the ITT Tech Virtual Library (http://library.itt-tech.edu/Pages/default.aspx/) to access online books, journals, and other reference resources selected to support ITT Tech curricula.

General References
- School of Study > School of Information Technology > Professional Organizations > Association of C & C++ Users
- School of Study > School of Information Technology > Recommended Links > Reference > C Programming: C++ Resources

Books
The following books are related to this course and are available through the ITT Tech Virtual Library
- eBrary > Easttom, William. *C++ Fundamentals: CyberRookies.* Charles River Media, 2003
- eBrary > Wright, Charles. *1001 Microsoft Visual C++ Programming Tips.* Course Technology, 2002
- NetLibrary > Horton, Ivor. *Beginning Visual C++.* US: John Wiley & Sons, Inc., 1998

Periodicals
- Periodicals > EbscoHost

All links to Web references outside of the ITT Tech Virtual Library are always subject to change without prior notice.

Evaluation & Grading

Course Requirements

1. **Attendance and Participation**
   Regular attendance and participation are essential for satisfactory progress in this course.

2. **Completed Assignments**
   Each student is responsible for completing all assignments on time.

3. **Team Participation (if applicable)**
   Each student is responsible for participating in team assignments and for completing the delegated task. Each team member must honestly evaluate the contributions by all members of their respective teams.

Evaluation Criteria Table

The final grade will be based on the following weighted categories:

CATEGORY	WEIGHT
Homework	15%
Lab	25%
Midterm Exam	15%
Project	25%
Final Exam	20%
Total	**100%**

Grade Conversion Table

Final grades will be calculated from the percentages earned in class 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

Course Outline

Notes

- **Unit 1**: All the concepts will be covered in the class; therefore, the specified readings are merely for your reference.
- **For all units, except Unit 1**: It is recommended that you complete the readings before attending the class.
- In addition to general class participation and in-class activities, all discussion questions will be graded under the Participation evaluation category.

Unit #	Activities for the unit
1—Introduction to C++ Programming	**Content Covered** **Chapter 1, "Introduction to Computers, the Internet and World Wide Web," pp. 6–11, pp. 14–23, and pp. 25–42** **Chapter 2, "Introduction to C++ Programming"** **Homework: 1** **Lab: 1.1**

Unit #	Activities for the unit

2—Controlling Execution	• **Content Covered**
	  ○ **Chapter 4, "Control Statements: Part 1," pp. 131–171, pp. 176–191**
	  ○ **Chapter 5, "Control Statements: Part 2"**
	• **Homework: 1**
	• **Lab: 2.1**
	• **Project Part 1: Start**

3—Functions	• **Content Covered** ○ **Chapter 6, "Functions and an Introduction to Recursion," pp. 244–304 and pp. 311–332** • **Homework: 1** • **Lab: 3.1** • **Project Part 2: Start** • **Project Part 1: Submit**

4—Classes and Objects—(Part 1)	• **Content Covered**
	  ○ **Chapter 3, "Introduction to Classes and Objects"**
	  ○ **Chapter 4, "Control Statements: Part 1," pp. 172–176**
	  ○ **Chapter 5, "Control Statements: Part 2," pp. 228–233**
	  ○ **Chapter 6, "Functions and an Introduction to Recursion," pp. 304–311**
	• **Homework: 1**
	• **Lab: 4.1**
	• **Project Part 3: Start**
	• **Project Part 2: Submit**

5—Classes and Objects—(Part 2)	• **Content Covered** ○ **Chapter 9, "Classes: A Deeper Look, Part 1"** • **Homework: 1** • **Lab: 5.1** • **Project Part 4: Start** • **Project Part 3: Submit**
6—Class Inheritance	• **Content Covered** ○ **Chapter 12, "Object-Oriented Programming: Inheritance"** • **Midterm Exam** • **Homework: 1** • **Lab: 6.1** • **Project Part 5: Start** • **Project Part 4: Submit**
7—Exception Handling	• **Content Covered** ○ **Chapter 16, "Exception Handling," pp. 817–832 and pp. 839–847** • **Homework: 1** • **Lab: 7.1** • **Project Part 6: Start** • **Project Part 5: Submit**
8—Arrays	• **Content Covered** ○ **Chapter 7, "Arrays and Vectors"** • **Homework: 1** • **Lab: 8.1** • **Project Part 7: Start** • **Project Part 6: Submit**

Unit #	Activities for the unit
9—Pointers and Strings	• **Content Covered** ○ **Chapter 8, "Pointers and Pointer-Based Strings"** • **Homework: 1** • **Lab: 9.1** • **Project Part 8: Start** • **Project Part 7: Submit**
10—Stream Processing	• **Content Covered** ○ **Chapter 15, "Stream Input/Output"** • **Homework: 1** • **Lab: 10.1** • **Project Part 9: Start** • **Project Part 8: Submit**
11—Review and Final Exam	• **Review and Final Exam** • **Project Part 9: Submit**