

Saint Leo University

COM 207 Programming in C/C++

Course Description:

The use of the C/C++ programming language for information processing and problem solving. Students use algorithms and computer logic to translate data into information through structured design, coding, testing, and program debugging.

Prerequisite:

COM 203

Textbooks:

Zak, D. (2016). *An introduction to programming with C++* (8th ed.). Boston, MA: Cengage Learning. ISBN-13: 978-1-285-86011-4

Materials:

Students will need to install Microsoft's Visual Studio Community 2015, which contains the Visual C++ compiler. You can download a free copy of Visual Studio Community from <https://www.visualstudio.com/en-us/downloads/visual-studio-2015-downloads-vs>

Learning Outcomes:

The student will:

1. Write, test, and debug C/C++ Code using the MS C/C++ IDE.
2. Develop algorithms using the sequence, select, and repetition structures; document the structure of these algorithms using flow charts and pseudo code; and translate these algorithms into procedural blocks (procedures, functions) of C/C++ code.
3. Create and use variables and named constants of the appropriate data types, including integers, floating point numbers, text strings, and one & two dimensional arrays; also convert data between different types as needed to process input and produce output.
4. Write C/C++ code to perform logical, arithmetic, and string operations on data.
5. Write code to interact with the user and file system using traditional I/O streams, file input/output, and also the dot net I/O techniques. In addition, this code will validate input and echo input values back to the user to ensure proper interpretation and prevent use of invalid results.
6. **VALUES OUTCOME:** In this class we will study how to organize data and effectively communicate information. Hence we will learn how to take precautions while recording data and presenting it in a manner that prevents misunderstanding, practicing Saint Leo University's core value of **Integrity**.

Integrity: The commitment of Saint Leo University to excellence demands that its members live its mission and deliver on its promise. The faculty, staff and students pledge to be honest, just and consistent in word and deed.

Grading:

The following criteria will be used to determine a final grade in the course:

Tests (Midterm & Final)	30%
Lab & Assignments:	60%
Weekly Discussion Questions:	10%

Grade/Score:

A	94-100%
A-	90-93
B+	87-89
B	84-86
B-	80-83
C+	77-79
C	74-76
C-	70-73
D+	67-69
D	60-66
F	0-59

Tests & Quizzes:

A midterm and final exam will be given and will cover the material presented since the last exam/test. The test given during finals week will NOT be a cumulative final. Make-up tests will be permitted only for excused absences, which would require written evidence.

Assignments:

Programming languages, just like human languages, must be learned by practice. Hence, this is a “hands on,” lab-intensive class. Students will be given specific graded assignments in each module throughout the course, all of which must be satisfactorily completed in order to complete the course. Computer assignments must be turned in electronically to the professor. All assignments and laboratory work for each chapter must be submitted according to the schedule. Late submission will result in the deduction of 10% off marks per day. No assignments will be accepted after three working days from the scheduled date and the student will be awarded zero marks for that part of the assignment. The assignment will be evaluated and the student notified of his grade, and the reasons for lost points.

Discussion Questions:

Discussion questions will be posted in each module. These represent an opportunity to interact with your classmates, and hopefully reinforce ideas, by allowing you to provide each other feedback. In each module you are expected to post your own reply to the question, and also respectfully comment on at least two (2) classmates' posts. You will be given a grade for the discussion in each module. Out of 100 points, an individual post is worth 70, and each comment on another's post is worth 10%.

Module 1 Getting Started

Objectives: **When you complete this module, you should be able to:**

- Define the terminology used in programming.
- Explain tasks performed by a programmer.
- Describe algorithms in terms of sequence, selection, and repetition structures.
- Explain the problem-solving process to create computer programs.
- Use IPO charts, pseudocode, and flowcharts to plan an algorithm.

Assignments:

Items to be Completed:	Due No Later Than
Read the assigned materials	
Post an introduction to the class	Thursday 11:59 PM EST/EDT
Post an initial response to the discussion question	Thursday 11:59 PM EST/EDT
Post responses to at least two classmates	Sunday 11:59 PM EST/EDT
Submit the Hands-On Learning assignment	Sunday 11:59 PM EST/EDT

Module 2 Variables, Constants, and Arithmetic

Objectives: **When you complete this module, you should be able to:**

- Distinguish among a variable, a named constant, and a literal constant.
- Explain how data is stored in memory, and identify fundamental data types in C++.
- Use C++ conventions to declare a memory location.
- Create variables of the appropriate data type and name them according to conventions.
- Use the assignment operator and arithmetic operators to perform basic computations.
- Use C++ stream operators to get user input from the keyboard and print output to the screen.

Assignments:

Items to be Completed:	Due No Later Than
Read the assigned materials	
Post an initial response to the discussion question	Thursday 11:59 PM EST/EDT
Post responses to at least two classmates	Sunday 11:59 PM EST/EDT
Submit the Hands-On Learning assignment	Sunday 11:59 PM EST/EDT

Module 3 Making Decisions

Objectives: **When you complete this module, you should be able to:**

- Use conditions and selection structures to allow programs to change their behavior based on the data they have received or the results of calculations.
- Use comparison operators and logical operators to make complex decisions.
- Code a selection structure using the **if** statement.
- Use nested and multiple-alternative selection structures in C++.
- Recognize common logical errors in C++.

Assignments:

Items to be Completed:	Due No Later Than
Read the assigned materials	
Post an initial response to the discussion question	Thursday 11:59 PM EST/EDT
Post responses to at least two classmates	Sunday 11:59 PM EST/EDT
Submit the Hands-On Learning assignment	Sunday 11:59 PM EST/EDT

Module 4 Going in Loops

Objectives: **When you complete this module, you should be able to:**

- Design pretest and posttest loops to accomplish computations.
- Compute totals and averages using accumulators and counters in loops.
- Code pretest loops using while statements and for statements.
- Code posttest loops using do while statements.
- Perform complex data manipulations using nested repetition structures.

Assignments:

Items to be Completed:	Due No Later Than
Read the assigned materials	
Post an initial response to the discussion question	Thursday 11:59 PM EST/EDT
Post responses to at least two classmates	Sunday 11:59 PM EST/EDT
Submit the Hands-On Learning assignment	Sunday 11:59 PM EST/EDT
Complete Midterm Exam	Sunday 11:59 PM EST/EDT

Module 5 Functions

Objectives: **When you complete this module, you should be able to:**

- Decompose larger algorithms into modular functions which can be created and tested in isolation.
- Classify activities into value-returning and void functions.
- Create function prototypes.
- Generate random numbers and timestamp using the system libraries.
- Pass parameters by reference or by value as needed.
- Create global and local variables as appropriate to the situation.

Assignments:

Items to be Completed:	Due No Later Than
Read the assigned materials	
Post an initial response to the discussion question	Thursday 11:59 PM EST/EDT
Post responses to at least two classmates	Sunday 11:59 PM EST/EDT
Submit the Hands-On Learning assignment	Sunday 11:59 PM EST/EDT

Module 6 Arrays

Objectives: **When you complete this module, you should be able to:**

- Create one- and two-dimensional arrays.
- Use loops to manipulate arrays.

Assignments:

Items to be Completed:	Due No Later Than
Read the assigned materials	
Post an initial response to the discussion question	Thursday 11:59 PM EST/EDT
Post responses to at least two classmates	Sunday 11:59 PM EST/EDT
Submit the Hands-On Learning assignment	Sunday 11:59 PM EST/EDT

Module 7 Strings

- Objectives:** **When you complete this module, you should be able to:**
- Create strings to hold text data.
 - Use the built-in string functions to manipulate and process strings.

Assignments:

Items to be Completed:	Due No Later Than
Read the assigned materials	
Post an initial response to the discussion question	Thursday 11:59 PM EST/EDT
Post responses to at least two classmates	Sunday 11:59 PM EST/EDT
Submit the Hands-On Learning assignment	Sunday 11:59 PM EST/EDT

Module 8 Text File Input and Output

- Objectives:** **When you complete this module, you should be able to:**
- Create files objects.
 - Open, read, and write data to a sequential access file.

Assignments:

Items to be Completed:	Due No Later Than
Read the assigned materials	
Post an initial response to the discussion question	Thursday 11:59 PM EST/EDT
Post responses to at least two classmates	Sunday 11:59 PM EST/EDT
Submit the Hands-On Learning assignment	Sunday 11:59 PM EST/EDT
Complete Final Exam	Sunday 11:59 PM EST/EDT