# COURSE OUTLINE

# ELN331

Prepared: Mark Allemang    Approved: Corey Meunier

| | |
|---|---|
| **Course Code: Title** | ELN331: C++ PROGRAMMING |
| **Program Number: Name** | 4029: ELECTRICAL TY-PROCES |
| **Department:** | ELECT./INSTRUMENTATION PS |
| **Semester/Term:** | 17F |
| **Course Description:** | This course will introduce students to high-level language programming. Assignments will include the application of high level language programming toward the solution of technical problems. Microsoft`s Visual C++ compiler will be used. |
| **Total Credits:** | 3 |
| **Hours/Week:** | 3 |
| **Total Hours:** | 45 |
| **Substitutes:** | OEL206 |
| **This course is a pre-requisite for:** | ELN321, ELN340 |
| **Vocational Learning Outcomes (VLO's):**<br>**Please refer to program web page for a complete listing of program outcomes where applicable.** | #8. Use computer skills and tools to solve a range of electrical related problems. |
| **Essential Employability Skills (EES):** | #1. Communicate clearly, concisely and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience.<br>#3. Execute mathematical operations accurately.<br>#4. Apply a systematic approach to solve problems.<br>#5. Use a variety of thinking skills to anticipate and solve problems.<br>#6. Locate, select, organize, and document information using appropriate technology and information systems.<br>#7. Analyze, evaluate, and apply relevant information from a variety of sources. |
| **Course Evaluation:** | Passing Grade: 50%, D |
| **Other Course Evaluation &** | The student will be expected to purchase an Arduino UNO microcontroller. |

Prepared: Mark Allemang Approved: Corey Meunier

**Assessment Requirements:**

The student must maintain a minimum 50% average in both the theory portion and lab portion of the class in order to receive a passing grade.

Grade
Definition Grade Point Equivalent
A+ 90 – 100% 4.00
A 80 – 89%
B 70 - 79% 3.00
C 60 - 69% 2.00
D 50 – 59% 1.00
F (Fail) 49% and below 0.00

CR (Credit) Credit for diploma requirements has been awarded.
S Satisfactory achievement in field /clinical placement or non-graded subject area.
U Unsatisfactory achievement in field/clinical placement or non-graded subject area.
X A temporary grade limited to situations with extenuating circumstances giving a student additional time to complete the requirements for a course.
NR Grade not reported to Registrar`s office.
W Student has withdrawn from the course without academic penalty.

**Evaluation Process and Grading System:**

| Evaluation Type | Evaluation Weight |
|---|---|
| Assignments and Quizes | 30% |
| Tests | 70% |

**Books and Required Resources:**

C++ A beginner’s Guide by Herbert Schildt
Publisher: McGraw, Hill/Osborne, Edition: 2nd Edition
ISBN: 978-0-07-223215-8, 0-07-223215-3

**Course Outcomes and Learning Objectives:**

## Course Outcome 1.

Describe and apply the concepts involved in the development of a program to solve problems using the computer and write simple C/C++ programs applying the concepts of input/output, arithmetic, and assignment.

## Learning Objectives 1.

define the concept of a computer program/software
differentiate between a high level language, compiler and machine language
describe the top-down process of developing a logical solution to a problem and use pseudocode to plan a series of detailed steps leading to a solution
demonstrate a basic understanding of the Microsoft Visual C++ environment
explain the main components of a C/C++ program
list and distinguish C/C++ basic data types
explain and properly use the naming conventions for C/C++ identifiers
differentiate between character, string, and numeric constants
differentiate between character and numeric variables
declare and initialize variables correctly
explain computer memory concepts and how they relate to processing data
use assignment operators (=, +=, -=, *=, /=) for assigning values/expression results to variables
use increment/decrement operators (++, --) to increase/decrease values by 1
use arithmetic operators and apply their precedence (+, -, *, /, %)
use various C++ math library functions to perform arithmetic calculations
describe the purpose of a compiler/interpreter
describe the process of transforming a source program to an executable module
differentiate between syntax and logic (semantic) errors
apply the cin >> object to perform input of data
apply the cout << object to perform output of data
apply the getline( ) function to accept string values that include a space(s)
explain and apply the #include directive
explain the purpose of include files for the cin and cout objects
write algorithms to solve problems using pseudocode
write, test, and debug programs to solve simple technical problems using the concepts above

## Course Outcome 2.

Develop algorithms and write C/C++ programs to solve problems involving the standard computer operations of decisions/conditions and selection.

## Learning Objectives 2.

describe and use the relational operators (==, !=, <, <=, >, >=)
describe the use of the logical operators (&&, ||) and use them to write both simple and complex expressions

describe the operation of the following C/C++ decision-making structures and use them in C/C++ programs:

if...else nested ifs if...else if...else the switch statement

write algorithms to solve problems containing decision-making structures, and describe them using pseudocode
write, test, and debug programs containing decision structures

## Course Outcome 3.

Develop algorithms and write C/C++ programs to solve problems involving the standard computer operations of looping and repetition, and, debug program logic errors using the C++ Debugger

## Learning Objectives 3.

discuss the concept of repetition/looping in computer programs
describe the operation of the following C/C++ repetition structures and use them in C/C++ programs:

while do...while for nested loops

use break, continue, and exit to terminate the iteration of a loop
write algorithms to solve problems containing repetition structures, and describe them using pseudocode
describe and correct an infinite loop problem
execute code one line at a time using the Step Debugger
define, as well as, insert and remove break point
write, test, and debug programs containing repetition structures

## Course Outcome 4.

Develop algorithms and write C++ programs to solve problems involving arrays.

## Learning Objectives 4.

define and apply the concepts of the following terms:

one-dimensional array , index value,
two-dimensional array , null character

discuss the purpose and concepts relating to one- and two-dimensional arrays
declare and initialize both numeric and character arrays
apply the concept of pointers to arrays
access and process array elements
pass arrays between functions
write, test, and debug programs containing arrays

## Course Outcome 5.

Discuss and apply the concepts of character sequences/arrays and string manipulation with reference to C/C++ library functions.

## Learning Objectives 5.

understand and utilize the C++ string class and its associated functions to declare string variables and manipulate string values
discuss and apply string functions such as:

strcat( ) strcmp( ) strlen( ) strcpy( )
atoi( ) atof( ) atol( ) itoa( )

write, test, and debug programs containing character and string functions

## Course Outcome 6.

Discuss and create user-written functions

## Learning Objectives 6.

understand the role and operation of functions in C/C++ and other languages
distinguish between the calling and the called functions

understand the concept of scope
distinguish between local and global variables
discuss and apply the concepts of ‘passing’ arguments to called functions by value, address and reference
discuss and apply the concept of ‘returning’ values to calling functions
write, test, and debug programs containing functions
discuss and apply the concept of pointers and pointer arithmetic
discuss and apply the concept of pointers in C/C++
define and apply the concepts of the following terms:

scope calling vs called functions function prototypes
local vs global variables pass by value return statement
arguments/parameters pass by reference

develop modularized, structured programs by creating user-written functions
discuss and apply the concepts of ‘passing’ arguments to called functions by value
discuss and apply the concept of ‘returning’ values to calling functions
discuss and apply the concepts of ‘passing’ arguments to called functions by reference
develop modularized, structured programs by creating user-written functions

## Course Outcome 7.

Build simple Arduino Microcontroller based applications.

## Learning Objectives 7.

Utilize the Arduino to perform digital input/output and analog input.
Write programs that utilize C++ programming language to perform simple I/O functions.

| | |
|---|---|
| **Date:** | Friday, September 1, 2017 |
| | Please refer to the course outline addendum on the Learning Management System for further information. |