AKENTEN APPIAH-MENKA
UNIVERSITY OF SKILLS TRAINING AND ENTREPRENEURIAL DEVELOPMENT

DEPARTMENT OF INFORMATION TECHNOLOGY EDUCATION

Course Code: ITC 127 Credit Hours: **3**

Course Description

This course provides in-depth coverage of object-oriented programming principles and techniques using C++. Topics include classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes, and low-level language features.

Course Objectives

- Perform object oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O. and other standard language constructs.
- Demonstrate adeptness of object oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.
- Demonstrate ability to implement one or more patterns involving realization of an abstract interface and utilization of polymorphism in the solution of problems which can take advantage of dynamic dispatching.
- Learn syntax, features of, and how to utilize the Standard Template Library. Learn other features of the C++ language including templates, exceptions, forms of casting, conversions, covering all features of the language. Learn features of the language which can be problematic with execution time or space and some techniques to resolve them.

Course Content

Unit 1: Introduction and Setup

This unit presents a brief history of C++ before addressing the mechanics of editing and compiling simple programs in C++ using the Eclipse IDE (integrated development environment). We will focus on how to write and format a general C++ program, the meaning of the `main()` function, how to use the `cout` and `cin` objects, how to declare and use variables, and how to use arithmetic operators.

Unit 2: Structuring Program Code

This unit focuses on implementing simple control structures. First, we will learn how to useconditional and iteration structures to make decisions and to repeat code. We will then discuss how to use debugging tools to test and troubleshoot these structures. We will also explore how to break our code into smaller, more manageable pieces by putting certain common pieces into functions. We will also discus scope, as well as passing variables by value and by reference. Finally, we explore a special type of class, the string, which has some special functions that allow us to manipulate text. By introducing classes and how they are used here, we will be ready to tackle object-oriented programming in the next unit.

Unit 3: Working with Simple Data Structures

Data structures are just ways to store multiple data values. Arrays, structs, enumerations, unions, queues, lists, and vectors are a few examples of data structures. In this unit, we will focus on a few simple structures. The array is one of the most basic structures used in computer programming. Arrays store data contiguously by representing data with a common name and distinguishing it by its index. This is like a parking garage: the garage stores vehicles; all of the parking spaces have a common name (the name of the garage); and each parked vehicle is identified by a specific parking space number. This is also true of arrays. Think of arrays as parking garages for our data. After exploring arrays, we then learn structs, unions, and enumerations, which are special ways to group more complex data types.

Unit 4: Object-Oriented Programming

In this unit, you will learn how to design a class, which is an expanded concept of a data structure that can hold both data and functions. An object is an instantiation of a class, so a class would be the type, and an object would be the variable. Next, we will learn how to handle private and protected members of a class, which is important for sound class design. This unit covers a key feature of C++ classes: inheritance. Inheritance allows classes to inherit objects and functions from other classes. In this unit, we will learn how classes can inherit members from more than one class. We will end this unit with the study of polymorphism or the ability to create a variable, a function, or an object that has more than one form. This brings object-oriented methodologies to their full potential.

Unit 5: Advanced Concepts

In this unit, we will first explore the concept of generic programming, and how you can harness the power of templates to make classes and functions more reusable and adaptable to your personal needs. Then, we will explore ways to add content to our program by reading from files and then storing the data by writing to output files. Finally, we will explore the use of exceptions in C++. Exceptions allow us to anticipate problems that might occur in code and handle this problems through the use of exceptions, telling the program specifically how it is to behave when these problems occur.

Method of assessment:

Assessment	Activities	Percentages (%)
Internal (40%)	Attendance Assignments (Lab Activities) Assessments (Quizzes)	5 15 20
Exam (60%)	Theory MCQ	20 40

Outline (weekly basis)

Week 1: Introduction to C++: First program, compilation, syntax errors

Week 2: Variables, data types, and arithmetic expressions

Week 3: Comparisons, simple conditions, ***if***, ***switch...case***

Week 4: More Data Types, conversions, constants, ***for*** loop, ***while*** loop

Week 5: Functions, parameter passing: pass by value, pass by reference

Week 6: Arrays. Array processing with ***for***

Week 7: Two-dimensional arrays

Week 8: Character arrays and string processing

Week 9: Pointers and pointer arithmetic

Week10: Object oriented programming

Week 11: Object oriented programming

Week 12: Introduction to Exceptional Handling

Week 13: Files

Week 14: Revision

Reading List

1. Ira Pohl, “Object Oriented Programming using C++”, 2nd Edition, PearsonEducation, Reprint 2004.

2. S. B. Lippman, JoseeLajoie, Barbara E. Moo, “C++ Primer”, 4th Edition,Pearson Education, 2005.
3. B. Stroustrup, “The C++ Programming Language”, 3rd Edition, PearsonEducation, 2004. Herbert Schildt, “C++: The Complete Reference”, TMH, 2006
4. B. Trivedi, “Programming with ANSI C++”, Oxford University Press, 2007.

Instructor:	Dr. William Asiedu
Phone:	+233247181354
Email:	asiedu2@gmail.com /wasiedu@uew.edu.gh

AMPOMAH PETER 4 INFOTESS