Course Outline

Unit Code & Name	CCS 2211: Object-Oriented Programming I
Prerequisite	CCS 2112: Introduction to Programming
Cohort	BCS & BSE Y2S1, Sept. – Dec. 2025
Lecturer	Dr. Muturi Peter
Contact	pmuturi@mmu.ac.ke

Purpose

To enable the student to understand the concepts and practices of object-oriented programming.

Learning outcomes

By the end of the course, the student should be able to:

1. Explain the principles of the object-oriented paradigm and its relationship to "traditional" methods.
2. Describe the process of object-oriented programming and design using the Object-Oriented language C++.
3. Demonstrate the use of object-oriented technologies and tools in problem-solving.
4. Develop an object-oriented application.

Course Description

Introduction to OOP basic concepts: classes, objects, inheritance, encapsulation, polymorphism, message passing; IDE for developing OOP programs; editing, compiling, and executing an OOP; General format of an OOP program; variables: local variables, global variables instance variables and static variables; arrays; single and multiple dimension arrays, data types, comments; abstract data types and primitive data type, typedef data types (classes): abstract/virtual and concrete classes. Implementation of OOP concepts: constructors and destructors: use of constructors and destructors, characteristics of a constructor. Inheritance of constructors. Functions: virtual, friend, pure virtual USE functions. Passing of values: pass-by-reference and pass-by-value. Errors and Exceptions: Thread and concurrency control. Use C++ for implementation.

Delivery Methodology

Lectures, laboratory exercises, assignments, and projects

Learning Resources

Books, Computers, Internet, Journals, Software required, Whiteboard, and Markers

Course Content

WEEK	TOPIC	OUTLINE
WK1	Introduction	Programming paradigms: Procedural to Object-Oriented.
WK2	Introduction to C++ and OOP	Overview of the C++ programming language Basics of OOP: Objects, classes, and methods Setting up a development environment
WK3	Introduction to C++ Language	C++ Syntax, Data types, and Control structures
WK4	Functions in C++	Inbuilt and user-defined functions. Parameter profile & passing
WK5	Classes and Objects	Defining classes Member functions and data members Constructors and destructors Access specifiers (public, private, protected)
WK6	Encapsulation and Abstraction **CAT1**	Encapsulation principles Using getter and setter methods

WK7	Class Instances and Message passing.	Class methods and objects: Constructors, destructors & other methods. Overloading.
WK8	Inheritance	Simple and Multiple inheritance
WK9	Polymorphism	Overloading and Overriding Virtual methods and pure virtual methods Abstract classes and interfaces
WK10	Exception Handling **CAT2**	Try-catch blocks Custom exception classes Handling runtime errors
WK11	File I/O and Streams	Reading from and writing to files File streams (ifstream, ofstream) Serialization and deserialization
WK12	Design Patterns	Creational patterns (e.g., Singleton, Factory) Structural patterns (e.g., Adapter, Decorator) Behavioral patterns (e.g., Observer, Strategy)
WK13	Project Work	Building a small-scale project using OOP principles Applying concepts learned throughout the course
WK14	Review and Final Assessment	Recap of key topics Project assessment presentation

Course Assessment

Continuous Assessment Tests	30%
End of Semester Examination	70%

Course Textbooks

1. Stroustrup, B. (2024). *Principles and Practice Using C++ (3rd Edition)*
2. Gaddis T. (2023). *Starting Out with C++ from Control Structures to Objects (10th Edition)*
3. Forouzan B. A. & Gilberg R. F. (2019). *C++ Programming: An Object-Oriented Approach*
4. Trivedi, B. (2007). *Programming with ANSI C++*. Oxford University Press.
5. ISRD Group (2007). *Introduction to OOP and C++*. Tata McGraw-Hill Publishing Company Ltd.
6. Malik, D.S. (2007). *C++ Programmimg: From Problem Analysis to Program Design* (3rd ed.). Thomson Course Technology.

Course Journals

1. International Journal of Software Engineering (IJSE)
2. Software – practice and Experience
3. International Journal of Advanced Software Engineering (IJASE)

Reference Textbooks

1. Bahrami, A. (2004). *Object –Oriented System Development*. McGraw-Hill Education.
2. Rambaugh, J. (1991). *Object Oriented Modelling and Design.* Prentice Hall.
3. Rajaram, R.(2006). *Object Oriented Programming and C++*(2nd 3ed.). New Age International.

Reference Journals

1. Web Developer's Journal
2. Knowledge and Information Systems
3. The Journal of Functional Programming