

**MULTIMEDIA UNIVERSITY OF KENYA** P.O. Box 15653 - 00503, Mbagathi, Nairobi Tel: +254 020 2071391, +254 020 724257083,

+254 020 735900008 Fax: +254 020 2071243Email:[**info@mmu.ac.ke**](mailto:info@mmu.ac.ke)

***Leader in Innovative Technology***

**FACULTY OF SCIENCE & TECHNOLOGY**

**DEPARTMENT OF MATHEMATICS**

**COURSE OUTLINE**

**UNIT CODE: ICS 2104: UNIT NAME: OBJECT ORIENTED PROGRAMMING I**

**Prerequisite:** CCS 2102 ~ INTRODUCTION TO COMPUTER PROGRAMMING

**LECTURER NAME:** MR. KELVIN KARIUKI

**CONTACTS:** 0729 385 557

**Twitter**: @teacherkaris

**Purpose:**

To introduce students to the concepts, skills; and practices of object oriented programming.

**Learning Outcomes:**

By the end of this unit students 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 an object oriented language
3. Demonstrate the use of object orientation technologies and tools in problem solving

**Course Description:**

Introduction to OOP basic concepts, 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 types, typed data types (classes): abstract/virtual and concrete classes. Implementation of OOP concepts: constructors and destructors: use of constructors and destructors, characteristics of a contractor. Inheritance of contractors. Function: 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.

**Content Summary**

|  |  |  |
| --- | --- | --- |
| **WEEK** | **TOPIC** | **SUB-TOPICS** |
| 1 | * **Introduction** | Programming paradigms: Procedural to Object Orientation. Object-Orientated Languages, Object Oriented Concepts: class, Object, Inheritance, Polymorphism. |
| 2 | * **Introduction to C++ Language** * **Lab 1** | History of C++, Differences between C and C++.  Data types, input/output handling, keywords, Operators and expressions. C++ programming environment. |
| 3 | * **Control structures** * **Lab 2** | Control Structures (1): Decision Making, *if/else* Statements, The *switch* Statement, Control Structures (2): Introduction, Iteration/looping, The *for* Statement, The *while* Statement, *Do-while* statement |
| 4 | * **Arrays, Strings and Pointers** * **Lab 3** | One-Dimensional array, Multidimensional arrays, string handling routines, Pointer arithmetic and manipulation: |
| 5 | * **Functions in C++** * **Lab 4** | Functions: User defined functions, Scope of Variables, Hiding variables, Parameters profile Pass by Value and Pass by Reference, Inline functions, Default Values/Arguments, inbuilt functions. |
| 6 | * **CAT1** |  |
| 7 | * **Object-Orientation** * **Lab 5** | Fundamentals of Classes: Class declarations, data members, member functions Encapsulation & Abstraction |
| 8 | * **Class Instances and Message passing.** | Creating objects, object message passing. |
| 9 | * **Memory management** * **Lab 6** | Constructors, destructors & other methods. Overloading constructors, array of objects, object pointers. |
| 10 | * **Inheritance** * **Lab 7** | Benefits of inheritance, Base and Derived classes. Implementing derived classes. Multiple inheritance, friend functions. |
| 11 | * **CAT2** * **Polymorphism** * **Lab 8** | Function overloading, Virtual functions, Pure virtual functions, functions overriding, Abstract classes. |
| 12 | * **Exceptions & Templates** | Exceptions & Error handling, Templates, Namespaces |
| 13 | * **File I/O in C++** * **Lab 9** | Writing to file, Reading from file |
| 14 | * **Project Presentation** | Programming project presentation |
| 15 & 16 | * **End of Semester Exams** |  |

**Course assessment:**

Assignments 10%

Laboratory reports 10%

Continuous Assessment Tests 10%

**Total Continuous Assessment 30%**

**End of Semester Examination 70%**

**Students Performance** will be assessed through continuous assessment tests, assignments, and laboratory practicals that will account for 30 %.

**Ground rules**

* 1. Late assignments will not be accepted
  2. Students must achieve 80% class attendance,
  3. Group work must be completed in time as required

**Practicals:**

**Laboratory 1:** Basic C++ Program Structure, Creating, Saving, Viewing and Editing a C++ project in CodeBlocks, Simple output. Input, variables and constants, operators.

**Laboratory 2:** If statements, switch, conditional operator. For, while and do while loops.

**Laboratory 3:** One-Dimensional array, Multidimensional arrays, string handling routines, Pointer arithmetic and manipulation.

**Laboratory 4:** Inline functions and Built-in functions.

**Laboratory 5:** Class declarations, data members, member functions Encapsulation & Abstraction.

**Laboratory 6:** Constructors, destructors & other methods. Overloading constructors, array of objects, object pointers.

**Laboratory 7:** Derived classes. Multiple inheritance, friend functions.

**Laboratory 8:** Function overloading, Virtual functions, pure virtual functions, functions overriding, abstract classes.

**Laboratory 9: C++** File handling.

**Teaching Methodology:** Lectures, practical; and tutorials.

**Instructional Materials**

1. LCD projector
2. Whiteboard
3. Computer

**Course Assessment:**

Continuous Assessment 30%

End of Semester Examination 70%

**Core Reading Materials:**

**Course Textbooks**

1. Trivedi,B. (2007) *Programming with ANSI C++.* Oxford University Press
2. ISRD Group (2007). *Introduction to OOP and C++.* Tata McGraw-Hill Publishing Company Ltd.
3. Malik, D.S. (2007).*C++ Programming*: From Problem Analysis to Program Design (3rd ed.). Thomson Course Technology.

**Course Journals**

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

**Reference Materials:**

**Reference Textbooks**

1. Bahrami, A. (2004). *Object –Oriented System Development.* McGraw-Hill Education
2. Rambaugh, J. (2005). *Object Oriented Modeling and Design.* Prentice Hall
3. Rajam, R. (2006). *Object Oriented Programming and C++* (2nd edition). New Age International.

**Reference Journals**

1. *Web Developer’s Journal. IEEE*
2. *Knowledge and Information Systems.* IEEE
3. *Journal of Functional Programming* IEEE

**Approved for use: Sign: (CoD)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Date**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Approved for use: Sign: (Dean, FoST)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Date**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_