



## MACHAKOS UNIVERSITY

### SCHOOL OF ENGINEERING & TECHNOLOGY DEPARTMENT OF COMPUTING & INFORMATION TECHNOLOGY

#### **Course Programme:**

Bsc Computer Science

#### **Course Unit:**

SCO 122: Structured Programming

#### **Course Unit Pre-requisite:**

SCO 120: programming methodology

#### **Semester & Academic Year**

First Year Second Semester 2025/2026

#### **Contact Hours & Credit Hours**

For Practical Unit: 42 Hours; 3 Hours Theory, 2 Hours Lab Work

#### **Lecturer:**

Grace Nyakorema

**Copyright © Machakos University, 2025 All  
Rights Reserved**

**MONTH, YEAR**

**January, 2025**

## **Course Purpose**

To familiarize the logic of structured programming approach and provide exposure in developing algorithm, flowchart and thereby writing efficient codes for user defined problem.

## **Expected Learning Outcomes of the Course**

At the end of this course unit, the student should be able to:

- i. Understand and explain the basic design principles for algorithms,
- ii. Understand core structured programming basics, principles, distinguish them and classify them.
- iii. Discuss the fundamental principles of Object– Oriented Programming, as well as in-depth data and information processing techniques
- iv. Know the C++ programming language and use it to write original code for problem solving,
- v. Know the tools for software development in C++ and use them to analyze problems, to construct solutions (algorithms) and to code them in C++,

## **Mode of Delivery**

Lectures, assignment, practical and tutorial sessions in Computer Laboratory, individual and group assignments, exercises and project work.

## **Instructional Materials and/or Equipment**

Overhead projector and computer, handouts, white boards, appropriate application software and internet.

## **Course Assessment**

Continuous Assessment (Tests 20%, Practical 20%) 40%

End of Semester Examination 60%

## Course Schedule/ Content

Week	Topic	Subtopic	Remarks/
1	Introduction to Programming	<ul style="list-style-type: none"> <li>• What is programming?</li> <li>• Programming paradigms</li> </ul>	Lecturer Discussion
2.	C++	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Standardization</li> <li>• Programming Environment</li> <li>• Basic Facilities</li> <li>• Computation</li> </ul>	Introduction Practical
3.	Variables	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Declaring a variable</li> <li>• Initiating a variable</li> <li>• Constants</li> <li>• Main function</li> <li>• Input and Output</li> </ul>	Reading Assignment
4	Selections and loops	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Selection</li> <li>• If statement</li> <li>• Comparison operators</li> <li>• Even or Odd</li> <li>• Else IF</li> <li>• Conditional Input</li> <li>• Switch statement Loops</li> <li>• While Loop</li> </ul>	Reading Assignment
5	CAT 1		

6	Arrays	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Declaring an Array</li> <li>• Initiating an Array</li> <li>• Copying an Array</li> <li>• Comparing Array</li> <li>• Two-dimensional Array</li> <li>• Sorting</li> <li>• Searching a sorted Array</li> </ul>	Group work
7	Strings	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Dta Type Char</li> <li>• Int char</li> <li>• Copying strings</li> <li>• Array with String Arrays</li> <li>• Sorting strings</li> </ul>	Lecturer Discussion

		<ul style="list-style-type: none"> <li>• Concatenating strings</li> </ul>	
8.	Functions	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Calling a function</li> <li>• Several return statements</li> <li>• Sub function</li> <li>• over ride functions</li> <li>• Header files</li> <li>• Recursive functions</li> </ul>	Reading Assignment
9.	Pointers	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Using pointers in C++</li> <li>• Null pointers</li> <li>• Pointer Arithmetic</li> <li>• Pointers vs Arrays</li> <li>• Pointer to Pointer</li> <li>• Return pointer from functions in C++</li> </ul>	Reading Assignment
10	<b>CAT 2</b>		
11	Data Structures	<ul style="list-style-type: none"> <li>• Defining a structure</li> <li>• Accessing structure members</li> <li>• Structures as function arguments</li> <li>• Pointers to structures</li> <li>• The typeded Keyword</li> </ul>	Lecturer Discussion

12	File Handling	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Opening a file</li> <li>• Closing a file</li> <li>• Writing to a file</li> <li>• Reading from a file</li> <li>• Read and Write examples</li> <li>• File position pointers</li> </ul>	Lecturer Discussion
13	Program Documentation	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Advantages of documentation</li> <li>• Examples</li> </ul>	Reading Assignment

		<ul style="list-style-type: none"> <li>• When to document</li> <li>• Information included on documentation</li> <li>• How to document.</li> </ul>	
14	<b>Revision</b>		

## Core Textbooks

- i. Bäckman, K. (2012). Structured Programming with C++. Bookboon.
- ii. Behrouz A. F, Richard F. G (2013), A Structured Programming Approach Using C, Cengage Learning 3rd Edition, ISBN– 10: 9780534491321
- iii. Tonny G. (2017), Starting out with C++: Early Objects. Addison– Wesley 9th Edition, ISBN– 13: 978– 0133360929
- iv. Bjarne S. (2013). The C++ Programming Language. Addison– Wesley, 4th Edition, ISBN– 13: 978– 0321563842

## Core Journals

- i. Minarova, N. (2013). Programming language paradigms & the main principles of object– oriented programming, 2, 77– 83
- ii. Helge E., Juan L., Frazer M., Ander I., (2018)CEAS Space Journal., Comparative study of programming languages for next– generation astrodynamics systems., 10., 115 – 123

## Recommended Reference Textbooks

- i. Philip D. (2018). Python 3 Object Oriented Programming Third Edition. ISBN 139781789615852
- ii. Introduction to Java Programming: Comprehensive Verison Tenth Edition, Pearson, ISBN 10: 0– 13– 376131– 2
- iii. Matt W. (2013). The Object– Oriented Thought Process. Addison– Wesley Professional 4th Edition (Developer’s Library), ISBN: 674– 7459765 Recommended

## Reference Journals

- i. Ben L., (2016) International Journal of Information and Communication Technology Research, C++ Code Structure and Algorithm Simulator., 6., 1– 8

- ii. Peter T., et.al., (2018) The Journal of Supercomputing., A taxonomy of task– based parallel programming technologies for high– performance computing., 74., 1422 – 1434.
- iii. Azad M. San Ahmed, R., M– Raouf Mahmood, S., M. Nabi, R., & L. Hussein, D. (2018). The Impact of Teaching Materials on Learning Computer Programming Languages in Kurdistan Region Universities and Institutes. *Kurdistan Journal of Applied Research*, 3(1), 27– 33.

### **Course Policy**

#### **1. Class Attendance**

Students are expected to attend all class sessions as indicated on the semester teaching timetable. A class attendance register will be used to tract the student's attendance. Note that if you don't attend 2/3 of the lectures, you will not be allowed to sit for end of semester examination

#### **2. Late Work Policy**

Be sure to pay close attention to deadlines. There will be no makeup assignments or CATs, or late work accepted without a serious and compelling reason and lecturer/Chairman of Department approval.

#### **3. Complete Assignments**

Assignments must be submitted by the given deadline or special permission must be requested from Lecturer before the due date. Extensions will not be given beyond the next assignment except under extreme circumstances.