

SYLLABUS

Module	C++ Programming	
Program	Computer science	
Faculty	Information Technology & Architecture	
Module code	CSC1203	
Lecturer	Name: mercy nyakundi Tel: 0787162273 E-mail: mmoraa@uok.ac.rw Consultation hours: 4pm-6pm Monday-friday	
Study cycle	Undergraduate	
Study trimester	I	
Module status	Major	
Amount of credits and distribution of hours	15 Credit hours	
	Lecture – 45hrs	Continuous assessment test – 2hrs
	Seminar - 3	Final evaluation -2hrs
Admission preconditions	Introduction to ICT and Programming in C	
Purposes of the module	In this module, students will learn basics of programming with C++, structured and Object Oriented Programming Concepts. They will learn how data abstraction, reusability, inheritance and modularity of code can be enhanced using C++.	
Learning outcomes	Upon completion of the course, students have the following general and professional knowledge, competences and abilities:	
Knowledge and understanding	Having successfully completed the module; students should be able to demonstrate knowledge and understanding of: <ol style="list-style-type: none"> i. Theoretical and practical programming concepts ii. Object Oriented concepts and their implementation in C++ iii. Basics data structures using C++. 	
Applying knowledge	Having successfully completed the module, students will be able to: Use various Programming language construct available in C++ Explain the principles of object oriented programming design Describe when and how to use the C++ statement like classes and functions	
Communication skills	Having successfully completed the module, students should be able to: <ol style="list-style-type: none"> i. Present their ideas to a general audience using reasonable written and oral communication skills. ii. Use numerical and statistical methods to solve Computational problems. 	

SYLLABUS

Learning skills	<p>Having successfully completed the module, students should be able to:</p> <ol style="list-style-type: none"> Demonstrate managerial skills Justify the importance of team spirit to solve a common problem. Be self contained to perform assigned tasks independently or with little guidance. Complete their assigned task under tough time constraints.
Content (the content of the teaching for each day will be outlined)	<p>Day 1: Overview of the course and syllabus. Introduction to object oriented programming using c++ programming Features of object oriented programming Research/discussion: The history and features of c++</p> <p>Day 2: C++ compilers/IDE. Structure of a c++ program. Creating, saving, compiling & linking and executing c++ program</p> <p>Day 3: C++ character set. Datatypes in c++ programming language. The identifiers and keywords. Declaring variables and assigning values to variables</p> <p>Day 4: The operators that can be used in c++ language Performing input and output operations</p> <p>Day 5: Decision making and branching statements The if statement The switch statement</p> <p>Day 6: Looping statements While and the do-while loop The for loop Jump statements; break, continue, go-to</p> <p>Day 7: Functions Introduction to user defined functions Functions with no arguments and no return value Function with arguments and no return value Function with arguments and return value Function with no arguments but returns value Function overloading Recursive function</p> <p>Day 8: Global and external variables Pointers</p>

SYLLABUS

	<p>Working with a single dimensional array Working with a two-dimentional array</p> <p>Day 9&10: Defining and accessing structures in C++ language Object oriented programming concepts:objects and classes Declaring classes</p> <p>Day 11: CATs</p> <p>Day 12: Access specifiers and class members Accessing the class members Class constructors and destructors</p> <p>Day 13 & 14: Inheritance and polymorphism</p> <p>Day 15: Review of the course</p> <p>Core Text</p> <p>Britton, Carol., [2005], <i>A student guide to object-oriented development</i>, Library Call No: 005.12 BRI 2005</p> <p>E. BalaguruSamy, [2004], <i>Object Oriented Programming with C++</i>, Second Edition, Tata McGraw Hill, New Delhi, ISBN 0-07-040211-6</p> <p>James Rumbaugh, Michael Blaha and others, [2002], <i>Object Oriented Modeling and Design</i>, PHI Pvt. Ltd, New Delhi, ISBN 81-203-1046-2</p> <p>Mark Allen Weiss., [2006], <i>Data Structures and Algorithm Analysis in C++</i>, Third Edition, Pearson International Edition, ISBN 0-321-39733-9</p> <p>Walter Savitch, [2007], <i>Problem Solving with C++</i>, Sixth Edition, Pearson International Edition, ISBN 0-321-44263-6</p> <p>Background Texts</p> <p>Mark A. Weiss. , [2005], <i>Data Structures and Problem Solving Using C++</i>, Third Edition, Addison-Wesley, ISBN 0-321-40992-2</p> <p>Paul J. Deitel, Harvey M. Deitel, [2005], <i>Simply C++ - An Application-Driven Tutorial Approach</i>, Prentice Hall, ISBN 0-13-127768-5</p> <p>Sara Baase, Allen Van Gelder , [2000], <i>Computer Algorithms-Introduction to Design and Analysis</i>, Third Edition, Addison Wesley, ISBN 0-201-61244-5</p>
Teaching / learning methods	Lecturer will introduce the concepts of structured programming. He/She will also describe some of the practical aspect of C programming. The Laboratories

SYLLABUS

assessments will be supervised by a tutor/lecturer/laboratory assistant. Students will try to work sometimes independently and in group to let them digest the problem and to own the understanding. During the lectures, there may be quizzes and discussions in groups with a plenary feedback. The corrections of laboratories exercises will be done in labs or in classes.

Teaching and learning process includes the following methods:

Lecture	<ul style="list-style-type: none"> • Presentation • Demonstration • Induction • Analysis • Case study • Teaching through electronic resources
Group work	<ul style="list-style-type: none"> • Discussion • Presentation • Demonstration • Case study • Synthesis • Research by using the internet
Practice / Lab work	<ul style="list-style-type: none"> • Demonstration • Individual work • Synthesis
Independent work	<ul style="list-style-type: none"> • Problem-based learning • Case study • Preparing presentations • Doing homework

Evaluation criteria

Assessment Strategy

Formative and summative assessments are organized.

In-course assessment composed of written test, assignment or homework and handled practical assignment must be organized. Students have to receive comments on their works and results where it is needed.

In-course assessment counts for 60% of the whole course marks while the final examination of 2 h 00' duration will count for 40%

Assessment Pattern

Component	Weighting (%)	Learning objectives covered
In-course assessment:	60 %	Objectives related to the part of the content to be assessed.
Final assessment:	40 %	Objectives related to the whole content.

SYLLABUS

--	--