


AMOU UNIVERSITY
“A Vehicle for Peace and Development”
AMOU UNIVERSITY



FACULTY OF COMPUTING AND ICT

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY PROGRAMME

ACADEMIC YEAR 2015/ 2016

COURSE DESCRIPTION

BIT 225	Programming Methodology(C)
Contact Hours	52 hours
Pre-requisite	BIT 211 IT Fundamentals
Purpose/Aim	To give students a thorough grounding in the key concepts, techniques, and methods that have emerged over time as programming has evolved into a process with increasing formalized approaches.
Course Objective (Indicative Learning Outcomes)	<p>Upon successful completion of this course unit, the learner should be able to demonstrate a satisfactory performance in the following areas:</p> <ul style="list-style-type: none"> ▪ Placing context development in programming languages and reasons for their evolutions ▪ Recognize and apply appropriate ways to control and manipulate data ▪ Break down and formulate data into structures appropriate for the problem to be addressed ▪ Define and apply a range of testing approaches to check the correctness of programming code ▪ Put programming activities into the context of software development as a whole with emphasis on the importance of documentation
Course Content	<ul style="list-style-type: none"> • History of Programming <p>Language models, evaluating languages, chronology of programming, language generations.</p> <ul style="list-style-type: none"> • Variables Controls Structures and calculations <p>Variable types and names, sequence, selection, loops.</p> <ul style="list-style-type: none"> • Data Analysis and Problems <p>Program development process: program specification, requirement analysis, design coding and analysis, implementation and support. structured diagrams, structured walkthrough</p> <ul style="list-style-type: none"> • Programming Techniques <p>Procedures and Functions, Arrays:- one dimensional arrays, two dimensional array, array processing.</p> <ul style="list-style-type: none"> • Testing <p>Why testing, levels of testing, unit, integration, systems, acceptance testing, and installation tests. Desk checking and dry running.</p> <ul style="list-style-type: none"> • Implementation <p>Traditional Systems lifecycle, SDLC, need for documentation and coding standards, attributes of a good documentation.</p>



AMOUD UNIVERSITY
“A Vehicle for Peace and Development”

	<p><u>Lab 1</u> Given a problem, come up with a pseudo-code to represent the solution to the problem. Draw flowcharts to represent solution of the problem above</p> <p><u>Lab 2</u> Given a problem, provide solution following the program development lifecycle i.e. Problem definition, design, coding, etc.</p> <p><u>Lab 3</u> Use Top-down problem-solving methodology to solve generic problems Introduction to concepts of structured programming e.g. Variables declaration, Control structures, statements, expressions</p> <p><u>Lab 4</u> Modularization to provide solution to complex problems including Functions and procedures</p> <p><u>Lab 5</u> Given a problem, design Inputs prompts as well as output screens</p> <p><u>Lab 6</u> Designing and analyzing algorithms, Mathematical foundations of algorithm design, Summations, Recurrences, Dynamic programming</p>		
Learning & Teaching Methodologies	Lectures, tutorials and computer laboratory exercises		
Instructional Materials/Equipment	Classroom with audio visual aids Computer laboratory High-level programming language e.g. C		
Course Assessment	Type		Weighting (%)
	Final Examination		60
	Mid Term Examinations		20
	Continuous Assessment(Assignments)		10
	Attendance		10
	Total		100
Recommended Reading	Title	Author	Publisher
	An introduction to Programming	Wu Thomas, Norman and Theodore	McGraw-Hill (1997)
Additional Reading	C Programming Language (2nd Edition)	Brian W. Kernighan, Dennis Ritchie, Dennis M. Ritchie	Prentice Hall PTR (2003)
	An introduction to algorithms	Cormen et al	MIT Press (2000)
Other Support Material	A variety of multimedia systems and electronic information resources as prescribed by the lecturer. Various application manuals, URL search and journals.		