

## Course: Programming logic and Design

<b>Department/ Programme: Computer Science (ND)</b>			
<b>Course: Programming logic and Design</b>	<b>Course Code:</b>	<b>Contact Hours:</b>	<b>5 hrs/week</b>
<b>Year: II Semester: I</b>	<b>Pre-requisite:</b>	<b>Theoretical:</b>	<b>2 hours/week</b>
		<b>Practical:</b>	<b>3 hours /week</b>
<b>General Objectives:</b>			
On completion of this course the student should be able to:			
1.0 Understand programming			
2.0 Understand Flowcharts and Pseudocode as Problem-Solving Modeling Tools.			
3.0 Understand Structured Programming: Decisions			
4.0 Know Modular Programmin			
5.0 Understand the simple for Loops			
6.0 Looping 2			
7.0 Algorithm			
8.0 Simple Data Structures			
9.0 Object-Oriented Programming			

g

Theoretical Content				Practical Content		
Week/s	Specific Learning Outcomes	Teacher's activities	Resources	Specific Learning Outcomes	Teacher's activities	Resources
<b>General Objective 1.0: Understanding Programming</b>						
1-3	Ability to understand:  Program Compilation and Execution  Program Statements, Programming Syntax  Variables and Data Types  C Operators:Arithmetic, Logic and Relational Operators  Simple I/O in C; the function main()	To: - Explain programming compilation and execution  - Explain Statements and Programming Syntax  - Explain Variables and Data Types  - Explain C Operators: Arithmetic, Logic and Relational Operators  - Simple I/O in C; the function main().	Projector to project slides. Markers	Write good programming	To assist students to do hands on of programming and know how to run a program	Personal computers loaded with an IDE.
<b>General Objective 2.0: Flowcharts and Pseudocode as Problem-Solving Modeling Tools</b>						
4-5	Ability to understand:  Structure of a C Program  Elements of a Flowchart  Simple Pseudocodes  Simple C Coding	To: -Explain the structure of a C program  - Explain the elements of a Flowchart.  -guide students to write simple C coding	Projector to project slides. Markers  Flowcharts do demonstrate problem solving model	Solve problems using flow charts	Assist students to write pseudocode to solve simple problems	A4 papers or books where they can write pseudocodes

<b>General Objective 3.0: Structured Programming: Decisions.</b>						
6-7	<p><b>Ability to understand:</b></p> <p>Simple C Decisions. The if-else Structure</p> <p>Logic Expressions</p> <p>Nested if-else</p> <p>The switch-case Structure</p> <p>and be able to write sample C program</p>	<p><b>To:</b></p> <ul style="list-style-type: none"> <li>- Explain the if else statement.</li> </ul> <p>Introduce students to writing logic Expression</p> <p>Explain nesting if-else</p> <p>Explain switch-case structure</p> <p>Introduce students to writing Sample C programming</p>	Whiteboard and marker and projector	Write simple programs to make decisions	To assist the students in writing and compiling simple decision making programs	Personal computers loaded with an IDE.
<b>General Objective 4.0: Modular Programming</b>						
8-9	<p>Ability to understand:</p> <p>Functions and Modular Programming</p> <p>C Examples of Problem-Solving Using Modules</p> <p>Function Arguments: Variables Passed to Functions</p> <p>Function Return Values: Variables Returned by Functions</p> <p>Function Prototype, definition and Call</p> <p>C Programming Examples</p>	<p><b>TO:</b></p> <ul style="list-style-type: none"> <li>- Explain functions and modular programming</li> <li>- List example of functions</li> <li>- Explain c examples in problem-solving using module</li> <li>Explain variable passes in function arguments</li> <li>- Explain function return values</li> <li>- Explain function return value: variable returned by functions</li> <li>Explain function prototype, definition and call</li> <li>Explain some c programming samples.</li> </ul>	Whiteboard and marker and projector and a laptop	Write modular programs	Assist students to write modular programs	Personal computers loaded with an IDE

<b>General Objective 5.0: Looping (I)</b>						
10 -12	<p>Ability to understand:</p> <p>Loops in Problem-Solving Examples</p> <p>The while Loop</p> <p>Counters, Control Variables, Loop Exit Conditions</p> <p>C Programming Examples Using the while Loop: Interactive Input,</p> <p>Data File-Reading and Writing</p>	<p>To:</p> <p>Explain how to solve problems using loops</p> <p>- Explain the while loop</p> <p>- Explain counters, control variable, loop and exit conditions</p> <p>-Give examples of each</p> <p>- Instructor goes through the c programming examples using loop: interactive inputs</p> <p>Explain the concept of data file- reading and writing</p>	<p>A flip chart.</p> <p>Whiteboard, marker and projector and a laptop</p>	<p>Write programs with loops</p>	<p>Guide students to run programs containing loops</p>	<p>Personal computers with IDE</p>
<b>General Objective 6.0: Looping (II)</b>						
13 -14	<p>Ability to understand:</p> <p>The do-while Loop</p> <p>C Programming Examples Using the do-while Loop</p> <p>Iterative Programming: The for Loop</p> <p>Incrementing, Decrementing for Loops</p> <p>C Programming Examples Using the for Loop</p>	<p>To:</p> <p>- Explain how the do-while loop work</p> <p>- Instructor goes through programming examples using the do- while loops</p> <p>Explain the iterative processes of the for loop</p> <p>- Explain the decreasing and increasing concepts</p> <p>-go through for looping examples</p>	<p>A flip chart.</p> <p>Whiteboard, marker and projector and a laptop</p>	<p>Write and run simple programs with do-while Loop</p>	<p>Assist students to write and run simple programs with loops</p>	<p>Personal computers Personal computers with IDE</p>
<b>General Objective 7.0: Algorithms</b>						
15-16	<p>Ability to understand:</p> <p>Search Algorithms (Sequential, Binary Search)</p> <p>Sort Algorithms (Insertion, Selection, File Indexing)</p> <p>C Program Examples</p>	<p>To:</p> <p>- Explain the search algorithms</p> <p>- Explain the sorting algorithms</p> <p>Go through c programming examples</p>	<p>Projector to project slides. Markers</p>	<p>demonstrate strong understanding of algorithmst</p>	<p>To assist students on their hand on practice</p>	<p>Personal computers loaded with an IDE.</p>

--	--	--	--	--	--	--

17-19	<b>General Objective 8.0: Simple Data Structures</b>					
	Ability to understand:	To:	Projector to project slides. Markers	Demonstrate strong understanding of the simple data structure	To assist students to do hands on of the simple data structures	Personal computers loaded with an IDE.
	Unions and Structures	- Explain Unions and Structures				
	Arrays of Structures	- Explain Arrays of Structures				
	Passing and Returning Structures	Explain passing and returning of structures				
	Abstract Data Types	Explain the Abstract Data Types				
	Problem Solving Using Structures	Lead students in solving problems using structures				
	Linked Lists	Explain Linked Lists				
	C Program Examples	Go through some c programs example				

20-22	<b>General Objective 9.0: Object-Oriented Programming</b>					
	Ability to understand:	To:	Projector to project slides. Markers and whiteboard	demonstrate good understanding of object-oriented programming	To assist students to do hands on of object-oriented programming	Personal computers loaded with an IDE.
	I/O in C++	- Explain the I/O in C++				
	Function Enhancements in C++	- Explain Enhancements in C++				
	Classes	Explain Classes				
	Class Constructors and Destructors	Explain Class Constructors and Destructors				
	C++ Program Examples	Demonstrate simple C++ program as example				

**Assessment:** Give details of assignments to be used: Coursework/ Assignments %; Course test 20 %; Practical 20 %; Projects %; Examination 60 %

Type of Assessment	Purpose and Nature of Assessment (COM 212)	Weighting (%)
Examination	Final Examination (written) to assess knowledge and understanding	60
Test	At least 2 progress tests for feedback ( students write the program on papers.	20
Practical	At least 5 home works to be assessed by the teacher	20
Total		100

**Recommended Textbooks & References:**

Farrell, Joyce: A Guide to Programming Logic and Design, Course Technology, ITP, 1999, ISBN 0-7600-1177-X

**Other recommended references:**

-course handouts