

ITEM	DETAILS
1. Title of subject	STRUCTURED PROGRAMMING
2. Subject code	STC103
3. Status of subject	Core
4. Stage	Year 1
5. Credit Hour	4
6. Pre-Requisite	None
7. Assessment	<p>40% Coursework Test 1 10% Test 2 10% Assignment 1 10% Assignment 1 10%</p> <p>60% Examination</p>
8. Semester	Semester 2
9. Objective of subject	<p>To enable students to:</p> <ul style="list-style-type: none"> Understand the key concepts, techniques and methods that have emerged over time as programming evolved into a process with increasingly formalized approaches. adapt to different software production situations with confidence and minimal adjustment to their understanding of the process using the skills and understanding developed in Structured Programming Methods.
10. Synopsis of subject	<p>Place in context developments in programming languages and reasons for their evolution; Recognize and apply appropriate ways to control and manipulate data; Break down and formulate data into structures appropriate for the problem to be addressed; Recognize and use a range of common programming concepts as building blocks in the creation of structured programs; Select and apply a range of techniques to modelling systems as objects and define code requirements from the models; Define and apply a range of testing approaches to check the correctness of programming code; Identify and explain other, evolving ways of designing and building programs; Put programming activities into the context of software development as a whole with emphasis on the importance of documentation.</p>

11. Details of subject	Contents	Hours
Week 1	Topic: 1. IMPORTANCE OF SOFTWARE DEVELOPMENT METHOD <ul style="list-style-type: none"> • Software Development Life Cycle • Requirement Specification • System Analysis • System Design • System Implementation • System Testing and Verification • Documentation and Maintenance 	4
	Learning Outcomes: At the end of the lessons, students will be able to: <ul style="list-style-type: none"> • Comprehend the software development life cycle and the importance of it. 	
Week 2 & 3	2. REPRESENTATION OF SYSTEM DESIGN Algorithm <ul style="list-style-type: none"> • Definition • importance of algorithm • different ways of writing algorithm • pseudo code and flowchart; • Variable Declaration; • Structure of Statements • Sequence structure, selection structure and repetition 	8
	Learning Outcomes: At the end of the lessons, students will be able to: <ul style="list-style-type: none"> • Understand algorithm and two ways of representing algorithm 	
	Activity: Tutorial and lab exercises .	
Week 4 & 5	3. FLOWCHART <ul style="list-style-type: none"> • Various symbols of representing flowchart • Advantages • Simple exercises using flowcharts • Comparison of pseudo code and flowchart 	Lecture 3 Tutorial 2 Lab 2
	Learning Outcomes: At the end of the lessons, students will be able to: <ul style="list-style-type: none"> • To know the symbols representing flowchart and how to draw the flowchart 	
	Activity: Tutorial and lab exercises	
	Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.	

<p>Week 6</p>	<p>4. IMPLEMENTATION – USING C++ AS PROGRAMMING TOOL</p> <ul style="list-style-type: none"> • Introduction to C++ programming language • Character set <ul style="list-style-type: none"> ○ Token ○ Reserved words ○ Identifiers ○ Constants ○ String literals ○ Punctuators <p>Learning Outcomes: At the end of the lessons, students will be able to:</p> <ul style="list-style-type: none"> • Be familiar with the character set of C++ <p>Activity: Tutorial and lab exercises</p> <p>Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.</p>	<p>Lecture 2 Tutorial 2 Lab 2</p>
<p>Week 7</p>	<p>5. OPERATORS</p> <ul style="list-style-type: none"> • unary, binary, bitwise • unary – increment and decrement operators, unary minus, size operator, cast operator • binary – arithmetic operators, assignment operators, arithmetic assignment operators, relational operators • bitwise - logical AND, logical OR • precedence of operators • library functions <p>Learning Outcomes: At the end of the lessons, students will be able to:</p> <ul style="list-style-type: none"> • Be familiar the different types of operators used in C++. <p>Activity: Tutorial and lab exercises</p>	<p>Lecture 2 Tutorial 2 Lab 2</p>
	<p>Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.</p>	
<p>Week 8</p>	<p>6. Structure of a C++ program</p> <ul style="list-style-type: none"> • Comments, preprocessor Directives, data type, type declaration, constants, statements, and compound statements <p>Learning Outcomes: At the end of the lessons, students will be able to:</p> <ul style="list-style-type: none"> • Understand the structure and able to write a C++ program <p>Activity: Tutorial and lab exercises</p>	<p>Lecture 2 Tutorial 2 Lab 2</p>
	<p>Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.</p>	

<p>Week 9</p>	<p>7. FORMATTED INPUT AND OUTPUTS</p> <ul style="list-style-type: none"> Input stream cin; output stream cout – formatting for simple and complex data; use of manipulators – endl, setw, setfill, setprecision and setiosflags. <p>Learning Outcomes: At the end of the lessons, students will be able to:</p> <ul style="list-style-type: none"> Comprehend the input and output formats in C++ 	<p>Lecture 2 Tutorial 2 Lab 2</p>
	<p>Activity: Tutorial and lab exercises</p> <p>Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.</p>	
<p>Week 10</p>	<p>8. REPETITION STATEMENTS</p> <ul style="list-style-type: none"> Importance of repetition statements; while loop, do-while loop; counter controlled and sentinel controlled looping 	<p>Lecture 2 Tutorial 2 Lab 2</p>
	<p>Learning Outcomes: At the end of the lessons, students will be able to:</p> <ul style="list-style-type: none"> Know the structure of repetition statements <p>Activity: Tutorial and lab exercises</p> <p>Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.</p>	
<p>Week 11</p>	<p>9. REPETITION STATEMENTS</p> <ul style="list-style-type: none"> For looping; Unconditional statements – goto; break; continue 	<p>Lecture 2 Tutorial 2 Lab 2</p>
	<p>Learning Outcomes: At the end of the lessons, students will be able to:</p> <ul style="list-style-type: none"> Know the structure of repetition statements <p>Activity: Tutorial and lab exercises</p> <p>Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.</p>	
<p>Week 12</p>	<p>10. STATEMENTS</p> <ul style="list-style-type: none"> sequence, selection , repetition; sequence statements – simple programs using the basic concepts; selection structure – single selection (if), double selection (if-else), multiple selection (switch – case) 	<p>Lecture 2 Tutorial 2 Lab 2</p>
	<p>Learning Outcomes: At the end of the lessons, students will be able to:</p> <ul style="list-style-type: none"> be familiar with the 3 types of statements and know in which situation to use them <p>Activity: Tutorial and lab exercises</p>	

	Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.	
Week 13	11. MODULAR PROGRAMMING APPROACH <ul style="list-style-type: none"> Importance of modular programming approach; Functions – components of a function – function definition – function call – function declaration 	Lecture 2 Tutorial 2 Lab 2
	Learning Outcomes: At the end of the lessons, students will be able to: <ul style="list-style-type: none"> Comprehend the importance of modular programming approach and functions 	
	Activity: Tutorial and lab exercises	
	Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.	
Week 14	12. PARAMETERS <ul style="list-style-type: none"> Actual and formal parameters – passing parameters using call by value and call by reference. Functions with return type, without return type; scope of a variable – local and global – visibility – duration of a variable Debugging – syntax errors, run-time errors, design errors, warning diagnostics errors 	Lecture 2 Tutorial 2 Lab 2
	Learning Outcomes: At the end of the lessons, students will be able to: <ul style="list-style-type: none"> Understand difference between local and global variables and the different types of errors. 	
	Activity: Tutorial and lab exercises	
	Further reading for this lesson: C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc.	
13. Text	Compulsory	1. C++ How to Program, 3/E, Harvey M. Deitel, Deitel & Associates, Inc. Paul J. Deitel, Deitel & Associates, Inc. © 2001 / 0-13-089571-7 / Prentice Hall
	Reference	1. Problem solving, Abstraction, Design using C++ Framl L.Friedman, Elliot B.Koffman Addison Wesley 2. Sams Teach Yourself C++ in 21 Days,4/E, Jesse Liberty © 2001 / 0-672-32072-X / Que/Sams 3. Schaum's Outline of Programming with C++ John Hubbard, Mc-Graw Hill.