

<b>1. Title of subject</b>	Computer Programming I
<b>2. Subject code</b>	TCP1231
<b>3. Status of subject</b>	Core Subject
<b>4. Credit hour</b>	3 28 Hours of Lecture 28 Hours of Lab LAN Credit Hours Equivalence: 3.00
<b>5. Semester</b>	Trimester 1 (Beta Level)
<b>6. Pre-Requisite</b>	None
<b>7. Methods of teaching</b>	28 Hours of Lecture 28 Hours of Lab
<b>8. Assessment</b>	60% Coursework 40% Final Exam <b>Total 100%</b>
<b>9. Teaching staff (Proposed)</b>	Ms. Emaliana
<b>10. Objective of subject</b>	To give an introduction to basic programming concepts through the use of a high-level programming language such as C/C++. It covers the basic notions and techniques for algorithm development and the implementation of algorithms in a high-level programming language.
<b>11. Synopsis of subject</b>	The major areas of study include: Software Development Life Cycle, Top-down Design, Program Design Steps and Programming Methodology, Structure Chart, Flowchart, Pseudo Code, Debugging and Documentation Techniques, Identifiers, Data Types, Operators, Various Statements, Type Conversion, Conditional and Control Structures, Functions, Arrays and Pointers, Strings, Structures and Unions, File Handling, Command Line Parameters, Pointers to Functions, Header Files, Stacks, Linked Lists, Bitwise Manipulation, and Programming laboratory exercises.

	Bidang pengajian meliputi: Kitaran hayat pembangunan perisian, Rekabentuk atas-bawah, Kaedah rekabentuk aturcara dan pengaturcaraan, Carta struktur, Carta alir, Kod pembayang, Teknik penyahpejatan dan dokumentasi, Pengenalpasti, Jenis-jenis data, Pengendali, Pelbagai kenyataan, Penukaran jenis, Struktur bersyarat dan struktur kawalan, Fungsi, Tatasusunan dan penunjuk, Rangkaian, Struktur dan gabungan, Penggunaan fail, Parameter baris perintah, Penunjuk ke fungsi, Fail pengepala, Timbunan, Senarai berpaut, Manipulasi bit, dan Latihan pengaturcaraan makmal.	
<b>12. Learning Outcomes</b>	By the end of the subject, students should be able to: <ul style="list-style-type: none"> <li>• Identify and apply basic concepts of a high level programming language correctly.</li> <li>• Demonstrate the basic notions and techniques for algorithm development</li> <li>• Implement these algorithms in a high-level programming language correctly and effectively.</li> </ul>	
	<b>Programmes Outcomes</b>	<b>Degree of contribution</b>
	Ability to apply soft skills in work and career related activities	<b>5</b>
	Good understanding of fundamental concepts	<b>40</b>
	Acquisition and mastery of knowledge in specialized area	<b>10</b>
	Acquisition of analytical capabilities and problem solving skills	<b>30</b>
	Adaptability and passion for learning	<b>5</b>
	Cultivation of innovative mind and development of entrepreneurial skills	<b>5</b>
	Understanding of the responsibility with moral and professional ethics	<b>5</b>
<b>13.Details of subject</b>	<b>Topics Covered</b>	<b>Hours</b>
1.	<b>Software Development and Programming Environment</b> Software development life cycle - Top-down design with function system structure - program design steps and programming methodology - structure chart - flowchart - pseudo code – Debugging and documentation techniques.	3
2.	<b>Syntactic Structure of a Program</b> Identifiers - data types – operators - various statements - operator precedence - type conversion – conditional and control structures - function - recursive functions.	4
3.	<b>Arrays</b> One- and two-dimensional arrays - passing individual elements or whole array to a function - Simple sorting and searching on arrays - pointers - strings - dynamic memory allocation.	6

	4.	<b>Structures and Unions</b> Structure declaration and definition – accessing structures - array of structures - pointers and structures – Union declaration – enumerated variables.	5
	5.	<b>File Handling</b> Concept of a file - files and streams – standard file handling functions - binary files - random access files.	5
	6.	<b>Advanced Topics</b> Command line parameters – pointers to functions - creation of header files - stacks - linked lists – bitwise manipulation	5
<b>Laboratory</b>		1. Students will be working programming exercises based on topics covered in the week.	
	<b>Total Contact Hours</b>		<b>28</b>
<b>14.Text</b>	<b>Text Book/Books</b>	1. Walter Savitch, <i>Problem Solving with C++</i> , Fourth Edition, Addison Wesley, 2004.	
	<b>Reference Book /Books</b>	1. Goran Svenk, <i>Object Oriented Programming using C++ for Engineering and Technology</i> , Thomson publishing, 2003. 2. Walter Savitch, <i>Problem Solving: The object of programming</i> , Fourth Edition, Addison Wesley, 2004. 3. D. S. Malik, <i>C++ programming</i> , Second Edition, Thomson Publishing, 2004. 4. Ira Pohl, <i>C++ by Dissection</i> , Addison Wesley, 2003.	