



NCEAC.FORM  
001-D

## COURSE DESCRIPTION FORM: CL-1002: Programming Fundamentals - Lab

**INSTITUTION** FAST School of Computing, National University of Computer and Emerging Sciences, Islamabad Campus

**BS-CS: Fall-2025**

### PROGRAM TO BE EVALUATED

#### Course Description

| <b>Lab Course Code</b>   | CL-1002   |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
|--|---|-----------------|--------|------------|-----------|------|----|-------------|---|----|--------------------------|---|----|------------|---|----|
| <b>Lab Course Title</b>  | Programming Fundamentals – Lab  |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>Credit Hours</b>  | 1   |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>Lab Course Instructors</b>  | Mr. Talha Shareef, Ms. Hajira Uzair, Ms. Zill-E-Huma  |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>Grading Policy</b>  | Absolute Grading  |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>Policy about missed assessment items in the course</b>  | Retake of missed assessment items (other than sessional/ final exam) will not be held. Students who miss an assessment item (other than sessional / final exam) would be awarded zero marks in that assessment item i.e. late submission will not be accepted. For missed sessional/ final exam, exam retake/ pretake application along with necessary evidence are required to be submitted to the department secretary. The examination assessment and retake committee decides the exam retake/ pretake cases. |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>Course Plagiarism Policy</b>  | Plagiarism in project or midterm/final exam will result in F grade in the course. Plagiarism in an assignment will result in zero marks in the whole assignment category.   |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>Prerequisites by Course(s) or Topics</b>  |   |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>Assessment Instruments with Weights</b><br>(Sessional exam, final exam, lab tasks, lab project, etc.) | <p>Assessment with the weight.</p> <table border="1"> <thead> <tr> <th>Assessment Item</th> <th>Number</th> <th>Weight (%)</th> </tr> </thead> <tbody> <tr> <td>Lab Tasks</td> <td>&gt;=14</td> <td>40</td> </tr> <tr> <td>Sessional-I</td> <td>1</td> <td>15</td> </tr> <tr> <td>Project (same as theory)</td> <td>1</td> <td>10</td> </tr> <tr> <td>Final Exam</td> <td>1</td> <td>35</td> </tr> </tbody> </table>  | Assessment Item | Number | Weight (%) | Lab Tasks | >=14 | 40 | Sessional-I | 1 | 15 | Project (same as theory) | 1 | 10 | Final Exam | 1 | 35 |
| Assessment Item  | Number  | Weight (%)      |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| Lab Tasks  | >=14  | 40              |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| Sessional-I  | 1   | 15              |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| Project (same as theory)   | 1   | 10              |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| Final Exam   | 1   | 35              |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>Lab Course Coordinator</b>  |   |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |
| <b>URL (if any)</b>  |   |                 |        |            |           |      |    |             |   |    |                          |   |    |            |   |    |



|                                     |   |                         |  |   |                      |   |   |                             |  |   |                                     |  |  |
|-------------------------------------|---|-------------------------|--|---|----------------------|---|---|-----------------------------|--|---|-------------------------------------|--|--|
| <b>Course Catalog Description</b>   | <p>The course aims to equip students with the basic computing concepts and to provide them the ability to analyze the given requirements for solving problems in different domains while implementing the solutions on a computer system. It emphasizes on developing an algorithm and applying the basic programming constructs like control structures, arrays, functions, pointers, dynamic memory allocation, etc. for its development. The students will learn the syntax of the C++ programming language for the implementation.</p>  |                         |  |   |                      |   |   |                             |  |   |                                     |  |  |
| <b>Laboratory Manual</b>            | Uploaded on LMS   |                         |  |   |                      |   |   |                             |  |   |                                     |  |  |
| <b>Course Goals</b>                 |   |                         |  |   |                      |   |   |                             |  |   |                                     |  |  |
|                                     | <p><b>A. Course Learning Outcomes (CLOs)</b></p> <p><i>After completion of the course, the students shall be able to:</i></p> <ul style="list-style-type: none"> <li>1. <b>Understand</b> basic problem-solving steps and logic constructs.</li> <li>2. <b>Apply</b> basic programming concepts.</li> <li>3. Design and implement algorithms to <b>solve</b> real-world problems.</li> </ul> <p><b>B. Program Learning Outcomes</b></p> <table border="1" data-bbox="535 952 1563 1886"> <tr> <td data-bbox="535 952 1432 1157">1. Computing Knowledge:</td> <td data-bbox="1432 952 1563 1157">Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solution of complex computing problems</td> <td data-bbox="1498 952 1530 1157">✓</td> </tr> <tr> <td data-bbox="535 1157 1432 1396">2. Problem Analysis:</td> <td data-bbox="1432 1157 1563 1396">Identify, formulate, research literature, and analyze complex computing problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, and computing sciences.</td> <td data-bbox="1498 1157 1530 1396">✓</td> </tr> <tr> <td data-bbox="535 1396 1432 1685">3 Design/Develop Solutions:</td> <td data-bbox="1432 1396 1563 1685">Design solutions for complex computing problems and design systems, components, and processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.</td> <td data-bbox="1498 1396 1530 1685">✓</td> </tr> <tr> <td data-bbox="535 1685 1432 1886">4. Investigation &amp; Experimentation:</td> <td data-bbox="1432 1685 1563 1886">Conduct investigation of complex computing problems using research based knowledge and research based methods.</td> <td data-bbox="1498 1685 1530 1886"></td> </tr> </table> | 1. Computing Knowledge: | Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solution of complex computing problems | ✓ | 2. Problem Analysis: | Identify, formulate, research literature, and analyze complex computing problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, and computing sciences. | ✓ | 3 Design/Develop Solutions: | Design solutions for complex computing problems and design systems, components, and processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations. | ✓ | 4. Investigation & Experimentation: | Conduct investigation of complex computing problems using research based knowledge and research based methods. |  |
| 1. Computing Knowledge:             | Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solution of complex computing problems  | ✓                       |  |   |                      |   |   |                             |  |   |                                     |  |  |
| 2. Problem Analysis:                | Identify, formulate, research literature, and analyze complex computing problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, and computing sciences.   | ✓                       |  |   |                      |   |   |                             |  |   |                                     |  |  |
| 3 Design/Develop Solutions:         | Design solutions for complex computing problems and design systems, components, and processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.  | ✓                       |  |   |                      |   |   |                             |  |   |                                     |  |  |
| 4. Investigation & Experimentation: | Conduct investigation of complex computing problems using research based knowledge and research based methods.  |                         |  |   |                      |   |   |                             |  |   |                                     |  |  |

|  |                                    |  |
|--|------------------------------------|--|
|  |                                    |  |
|  | 5. Modern Tool Usage:              | Create, select, and apply appropriate techniques, resources and modern computing tools, including prediction and modeling for complex computing problems.              |
|  | 6. Society Responsibility:         | Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal, and cultural issues relevant to the context of complex computing problems. |
|  | 7. Environment and Sustainability  | Understand and evaluate sustainability and impact of professional computing work in the solution of complex computing problems.  |
|  | 8. Ethics                          | Apply ethical principles and commit to professional ethics and responsibilities and norms of computing practice.   |
|  | 9. Individual and Teamwork:        | Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.  |
|  | 10. Communication                  | Communicate effectively on complex computing activities with the computing community and with society at large   |
|  | 11. Project Management and Finance | Demonstrate knowledge and understanding of management principles and economic decision making and apply these to one's own work as a member of a team.                 |
|  | 12. Lifelong Learning              | Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological changes.         |



| <b>Topics covered in the course</b><br>(assume 15-week instruction and 3 contact hours of lab per week) | <b>C. Mapping of CLOs on PLOs</b><br>(CLO: Course Learning Outcome, PLOs: Program Learning Outcomes)  |              |               |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|--------------|---------------|--------|---|---|---|---|---|---|----|--------|--------|----------------|--------------|---------------|--------|-------------------------------------|---|---|---|-------------------------|---|---|-----|--|---|---|-----|------------------------|---|---|-----|--------------------------------------|---|---|-----|--|---|---|-----|---|---|---|-----|--------------------------------------|---|---|-----|--|---|---|-----|--|---|---|-----|------------------|--|--|--|--|--|--|--|--|--|--|--|--|---|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
|   | <b>PLOs</b>   |              |               |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   |   | 1            | 2             | 3      | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1<br>1 | 1<br>2 |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   | <b>C<br/>L<br/>O<br/>s</b>  | 1            | ✓             | ✓      |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   |   | 2            |               | ✓      | ✓ |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   |   | 3            |               |        | ✓ |   |   |   |   |   | ✓  |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   |   |              |               |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   |   |              |               |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   | <b>1. Topics to be covered:</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 5px;">List of Topics</th> <th style="text-align: center; padding: 5px;">No. of Weeks</th> <th style="text-align: center; padding: 5px;">Contact Hours</th> <th style="text-align: center; padding: 5px;">CLO(s)</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">Ubuntu installation, Shell commands</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1</td> </tr> <tr> <td style="padding: 5px;">Pseudo Code and Scratch</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td style="padding: 5px;">Basic program writing in C++ and stream insertion/extraction operators and datatypes</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td style="padding: 5px;">Operators (arithmetic)</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td style="padding: 5px;">Operators (arithmetic, Bitwise etc.)</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td style="padding: 5px;">Conditional structures-I (if-else, ternary operator)</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td style="padding: 5px;">Conditional structures-II (switch-case)</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td style="padding: 5px;">Repetitions-I (while loop, for loop)</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td style="padding: 5px;">Repetitions-II (do-while loop, nested while loop, nested for loop)</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td style="padding: 5px;">Repetitions-II (do-while loop, nested while loop, nested for loop)</td> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">3</td> <td style="text-align: center; padding: 5px;">1,2</td> </tr> <tr> <td colspan="13" style="text-align: center;"><b>Sessional</b></td> </tr> <tr> <td colspan="13">           Arrays-I (1D arrays)      1      3      1,2         </td> </tr> <tr> <td colspan="13">           Arrays-II (char and multi-dimensional arrays)      1      3      1,2         </td> </tr> </tbody> </table> |              |               |        |   |   |   |   |   |   |    |        |        | List of Topics | No. of Weeks | Contact Hours | CLO(s) | Ubuntu installation, Shell commands | 1 | 3 | 1 | Pseudo Code and Scratch | 1 | 3 | 1,2 | Basic program writing in C++ and stream insertion/extraction operators and datatypes | 1 | 3 | 1,2 | Operators (arithmetic) | 1 | 3 | 1,2 | Operators (arithmetic, Bitwise etc.) | 1 | 3 | 1,2 | Conditional structures-I (if-else, ternary operator) | 1 | 3 | 1,2 | Conditional structures-II (switch-case) | 1 | 3 | 1,2 | Repetitions-I (while loop, for loop) | 1 | 3 | 1,2 | Repetitions-II (do-while loop, nested while loop, nested for loop) | 1 | 3 | 1,2 | Repetitions-II (do-while loop, nested while loop, nested for loop) | 1 | 3 | 1,2 | <b>Sessional</b> |  |  |  |  |  |  |  |  |  |  |  |  | Arrays-I (1D arrays)      1      3      1,2 |  |  |  |  |  |  |  |  |  |  |  |  | Arrays-II (char and multi-dimensional arrays)      1      3      1,2 |  |  |  |  |  |  |  |  |  |  |  |  |
|   | List of Topics  | No. of Weeks | Contact Hours | CLO(s) |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ubuntu installation, Shell commands   | 1   | 3            | 1             |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pseudo Code and Scratch   | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Basic program writing in C++ and stream insertion/extraction operators and datatypes                    | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Operators (arithmetic)  | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Operators (arithmetic, Bitwise etc.)  | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Conditional structures-I (if-else, ternary operator)  | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Conditional structures-II (switch-case)   | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Repetitions-I (while loop, for loop)  | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Repetitions-II (do-while loop, nested while loop, nested for loop)                                      | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Repetitions-II (do-while loop, nested while loop, nested for loop)                                      | 1   | 3            | 1,2           |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <b>Sessional</b>  |   |              |               |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Arrays-I (1D arrays)      1      3      1,2   |   |              |               |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Arrays-II (char and multi-dimensional arrays)      1      3      1,2                                    |   |              |               |        |   |   |   |   |   |   |    |        |        |                |              |               |        |                                     |   |   |   |                         |   |   |     |  |   |   |     |                        |   |   |     |                                      |   |   |     |  |   |   |     |   |   |   |     |                                      |   |   |     |  |   |   |     |  |   |   |     |                  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



|   |   |                                      |                       |                                  |   |
|---|---|--------------------------------------|-----------------------|----------------------------------|---|
|   | Functions (parameter passing by value/reference) and Recursion  | 1                                    | 3                     | 1,2,3                            |   |
|   | Introduction to pointers and dynamic memory allocation (for 1D) | 1                                    | 3                     | 1,2,3                            |   |
|   | 2D and 3D   | 1                                    | 3                     | 1,2,3                            |   |
|   | Total   | 15                                   | 45                    |                                  |   |
| <b>Practical/ Programming Work/ Tools</b> | Ubuntu, Ubuntu shell, Text Editor, g++                          |                                      |                       |                                  |   |
| <b>Lab Time Spent (in percentage)</b>     | <b>Theory</b>   | <b>Problem Analysis &amp; Design</b> | <b>Implementation</b> | <b>Social and Ethical Issues</b> |   |
|   | 5   | 15                                   | 70                    |                                  | 2 |