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

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

BS-SE: **Fall-2022**

**PROGRAM TO BE EVALUATED**

**Course Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lab Course Code** | CL-1002 | | | |
| **Lab Course Title** | Programming Fundamentals - Lab | | | |
| **Credit Hours** | 1 | | | |
| **Lab Course Instructors** | Muhammad Hanan, Zoya Sumbul Zaheer | | | |
| **Grading Policy** | Absolute Grading | | | |
| **Policy about missed assessment items in the course** | Retake of missed assessment items (other than sessional/ final exam) will not be held. Student who misses an assessment item (other than sessional / final exam) is 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 decide the exam retake/ pretake cases. | | | |
| **Course Plagiarism Policy** | 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. | | | |
| **Prerequisites by Course(s) or Topics** |  | | | |
| **Assessment Instruments with Weights**  (Sessional exam, final exam, lab tasks, lab project, etc.) | Assessment with the weight.   |  |  |  | | --- | --- | --- | | **Assessment Item** | **Number** | **Weight (%)** | | Lab Tasks | 15 | 40 | | Assignments | 5 | 05 | | Midterm Exam | 1 | 20 | | Project | 1 | 05 | | Final Exam | 1 | 30 | | | | |
| **Lab Course Coordinator** | Muhammad Hanan | | | |
| **URL (if any)** |  | | | |
| **Course Catalog Description** | 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. | | | |
| **Laboratory Manual** | Uploaded on LMS | | | |
| **Course Goals** | |  | | --- | | **A. Course Learning Outcomes (CLOs)** | | After completion of the course, the students shall be able to:   1. **Understand** basic problem-solving steps and logic constructs. 2. **Apply** basic programming concepts. 3. Design and implement algorithms to **solve** real-world problems. | | |  |  | | --- | --- | | **B. Program Learning Outcomes** | | | |  |  | | --- | --- | | 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 analyse 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 modelling for complex computing problems. | |  | | |  |  | | --- | --- | | 6. Society Responsibility: | Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal, and cultural issues relevant to 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. | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **C. Mapping of CLOs on PLOs**  (CLO: Course Learning Outcome, PLOs: Program Learning Outcomes) | | | | | | | | | | | |  |  | |  | | **PLOs** | | | | | | | | | |  |  | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | | **CLOs** | 1 | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  | | 2 |  | ✔ | ✔ |  |  |  |  |  |  |  |  |  | | 3 |  |  | ✔ |  |  |  |  |  | ✔ |  |  |  | | | | | |
| **Topics covered in the course**  (assume 15-week instruction and 3 contact hours of lab per week) | |  |  |  |  | | --- | --- | --- | --- | | **1. Topics to be covered:** | | | | | List of Topics | No. of Weeks | Contact Hours | CLO(s) | | Ubuntu installation, Shell commands | 1 | 3 |  | | Introduction to pseudo-code, algorithms, and flow chart with scratch | 1 | 3 |  | | Basic program writing in C++ and stream insertion/extraction operators | 1 | 3 | Assignment-1 (pseudocode, output manipulation) | | Operators (arithmetic, bitwise etc.) | 1 | 3 |  | | Functions (definition, calling, forward declaration) | 1 | 3 | Assignment 2 (Expression evaluation, operators) | | Conditional structures-I (if-else, ternary operator) | 1 | 3 |  | | Conditional structures-II (switch-case) | 1 | 3 | Assignment-3 (Control Structures) | | Repetitions-I (while loop, for loop) | 1 | 3 |  | | Repetitions-II (do-while loop, nested while loop, nested for loop) | 1 | 3 | Assignment-4 (Repetitions) | | Arrays-I (1D arrays) | 1 | 3 |  | | Arrays-II (char and multi-dimensional arrays) | 1 | 3 | Assignment-5 (Arrays) | | Functions (parameter passing by value/reference) | 1 | 3 | Project | | Basic File Handling | 1 | 3 |  | | Introduction to pointers and dynamic memory allocation (for 1D) | 1 | 3 |  | | Dynamic memory allocation (2D, and 3D) | 1 | 3 |  | | Total | **15** | **45** |  | | | | |
| **Practical/ Programming Work/ Tools** | Scratch,Ubuntu, Ubuntu shell, g++ | | | |
| **Lab Time Spent**  (in percentage) | **Theory** | **Problem Analysis & Design** | **Implementation** | **Social and Ethical Issues** |
| 5 | 15 | 78 | 2 |