**COURSE DESCRIPTION FORM:** **CS-1004: Object-Oriented Programming**

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

BS-CSDF: **Spring-2023**

**PROGRAM(s) TO BE EVALUATED**

**Course Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Code** | CS-1004 | | | |
| **Course Title** | Object-Oriented Programming | | | |
| **Credit Hours** | 3+1 | | | |
| **Course Instructors** | Zainab Abaid, Jawad Hassan | | | |
| **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 decides the exam retake/ pretake cases. | | | |
| **Course Plagiarism Policy** | TBA. | | | |
| **Prerequisites by Course(s) or Topics** | Programming Fundamentals (CS-1002) | | | |
| **Assessment Instruments with Weights** (homework, quizzes, sessional exams, final exam, assignments, etc.) | Assessment items of theory part with the weight:   |  |  |  | | --- | --- | --- | | **Assessment Type** | **Number** | **Weight** | | Assignments | 3 | 15 | | Quizzes | 4 | 10 | | Sessional-I | 1 | 12.5 | | Sessional-II | 1 | 12.5 | | Project | 1 | 10 | | Final Exam | 1 | 40 |   Assessment items of lab part with the weight:   |  |  |  | | --- | --- | --- | | **Assessment Type** | **Number** | **Weight** | | Lab Tasks | 15 | 15 | | Assignments | 5 | 10 | | Midterm Exam | 1 | 20 | | Project | 1 | 15 | | Final Exam | 1 | 40 | | | | |
| **Course Coordinator** | Zainab Abaid | | | |
| **URL (if any)** |  | | | |
| **Course Catalog Description** | Object-oriented programming principles; memory management; pointers; recursion; structures; interface vs. implementation; objects; classes; function types; macros; function overloading; operator overloading; function overloading; composition; association; aggregation; inheritance; function overriding; polymorphism; virtual vs. non-virtual functions; abstract vs. concrete classes; file-handling; templates | | | |
| **Textbook(s)** | Tony Gaddis “Starting Out with C++ from Control Structures to Objects” 8th Edition | | | |
| **Reference Material** | Paul Deitel, Harvey Deitel "C++ How to Program" 10th Edition  Walter Savitch "Problem Solving with C++" 10th Edition  D. S. Malik "C++ Programming: From Problem Analysis to Program Design" 8th Edition | | | |
| **Course Goals** | |  | | --- | | **A. Course Learning Outcomes (CLOs)** | | After completion of the course, the students shall be able to:   1. **Demonstrate** the basic concepts of OOP 2. **Apply** OOP concepts (Encapsulation, Inheritance, Polymorphism, Abstraction) to computing problems for the related program 3. **Model** an algorithmic solution for a given problem using OOP 4. **Apply** good programming practices. | | **B. Program Learning Outcomes (PLOs)** | | |  |  |  | | --- | --- | --- | | **PLO 1** | Computing and Cybersecurity Knowledge | Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solve complex problems using computer networks and cybersecurity techniques | | **PLO 2** | Problem Analysis | Identify, formulate, research literature, and analyze complex computational problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, computing, and cybersecurity | | **PLO 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. | | **PLO 4** | Investigation & Experimentation | Conduct investigation of complex computing and security problems using research-based knowledge and research-based methods | | **PLO 5** | Modern Tool Usage | Create, select, and apply appropriate techniques, resources and modern computing and cybersecurity tools for complex security problems | | **PLO 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 | | **PLO 7** | Environment and Sustainability | Understand and evaluate sustainability and impact of professional computing and cybersecurity work in solving complex computing and security problems | | **PLO 8** | Ethics | Apply ethical principles and commit to professional ethics and responsibilities and norms of computing and cybersecurity practice | | **PLO 9** | Individual and Team Work | Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings | | **PLO 10** | Communication | Communicate effectively on complex computing and cybersecurity activities with the cybersecurity community and with society at large | | **PLO 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 or a team | | **PLO 12** | Life Long 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 to 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 |  |  | ✓ | ✓ | ✓ |  |  |  | ✓ | ✓ | ✓ |  | |  | 4 | ✓ |  | ✓ |  |  |  |  |  |  |  |  | ✓ | | |  | | | | |
| **Topics covered in the course**  (assume 15-week instruction and 3 contact hours per week) | |  |  |  |  | | --- | --- | --- | --- | | **1. Topics to be covered:** | | | | | List of Topics | No. of Weeks | Contact Hours | CLO(s) | | Memory Layout, pointers, dynamic memory allocation, stack vs. heap, pointer arithmetic, pointer vs. array, multidimensional pointers, pointers types, char\* pointers, alias to pointers | 1.33 | 4 | 2, 4 | | Recursion, recursion for patterns and problem solving | 1 | 3 | 2,3,4 | | Introduction to object-oriented design, structures (struct keyword), access specifiers, member functions | 1 | 3 | 2,4 | | Introduction to classes (class keyword), constructors vs. destructors, copy constructor, overloading constructors, this pointer for function call resolution | 1.33 | 4 | 2,4 | | Inline functions vs. macros, constant vs non-constant functions, static data members and functions | 1 | 3 | 2,4 | | Function overloading, introduction to operator overloading, unary and binary operators overloading, member vs. global functions for operator overloading, stream insertion and extraction operators overloading | 2.33 | 7 | 2,4 | | Identifying classes and defining relationships, introduction to composition (Association & Aggregation), separating interface from implementation | 0.66 | 2 | 2,4 | | Introduction to inheritance, types of inheritance, function overriding, function overriding vs. overloading, single vs.  multiple inheritance | 1 | 3 | 2,4 | | Introduction to polymorphism, virtual vs.  non-virtual functions, abstract vs.  concrete classes | 2.66 | 8 | 2,4 | | Introduction to streams and file handling, files modes, ASCII vs binary file handling, sequential vs. random file reading and writing | 1.33 | 4 | 2,3,4 | | Introduction to templates, template functions and template classes, C++-11 auto keyword | 1.33 | 4 | 2,3,4 | | Total | **15** | **45** |  | | | | |
| **Programming Language for Assignments (if any)** | C++ | | | |
| **Class Time Spent**  (in percentage) | **Theory** | **Problem Analysis** | **Solution Design** | **Social and Ethical Issues** |
| 55 | 30 | 10 | 5 |
| **Oral and Written Communications** | Every student is required to submit at least \_\_3\_\_\_ coding assignments and \_\_\_\_1\_\_\_ coding project with a written report of approximately \_\_\_8\_\_\_ pages. | | | |