

CS1004 - Object Oriented Programming

| Form number | COURSEWARE OUTLINE/ DOCUMENT (Tentative) | | | |
|-------------------------------|--|--|------------|----------------------------|
| COURSE INSTRUCTOR INFORMATION | Name | Engr. Muhammad Usman Malik | | |
| | email ID | usman.malik@nu.edu.pk | | |
| | Contact | ----- | | Computer Engineering Dept. |
| DEGREE INFORMATION | Program | Batch | Section(s) | Semester |
| | BS(CS) | 2024 | BSCS-2A | Spring Year 2025 |

| | | | | | |
|--------------------|--|--------------|---|-----------------------------|--------------|
| COURSE INFORMATION | Course Category C- Core/ E-Elective | | Code | Title | Credit hours |
| | C | | CS1004 | Object Oriented Programming | 3+1 |
| | Prerequisite(s) | | CS 118 | Programming Fundamentals | 3+1 |
| | TA Required (Yes/ No) | No. of TA(s) | Brief Justification | | |
| | Yes | 3 | For assignments, tutorials, and improvised coordination | | |

| | | | |
|--------------------------|----------------------|---------------|--|
| TEXT BOOK(S) INFORMATION | Title of Book | | Object-Oriented Programming in C++ (Robert Lafore) C++ How to program (Deitel & Deitel) |
| Reference Book (s) | 1. | Title of Book | C++ Programming: From Problem Analysis to Program Design (D.S. Malik) |
| | 3. | Title of Book | The C++ Programming Language (Bjarne Stroustrup) |
| | Support Material (s) | a. | |
| | | b. | |

| | |
|---|--|
| Brief Description of Course: (not more than 250 words) | This is an advanced course on programming where the emphasis would be on programming skills so that students would be able to write a program of reasonable size and complexity and handle more complex computational applications and more importantly get introduced to the basic concepts of object-oriented programming. |
|---|--|

| Course Objectives (CO): | |
|-------------------------|---|
| 1. | Critical Analysis: The course aims to provide students the ability to analyze the given requirements for solving simple problems that can be implemented on the computer system. |
| 2. | Solution finding: The computer-programming course attempts to teach students the art of designing algorithm-based solutions to solve problems in different domains. |
| 3. | Interface and Implementation: The course would teach students the syntax and control structures of a programming language to implement algorithms for solving a particular problem. It would emphasize on the need to separate the interface from the implementation. |
| 4. | Collaboration: The course stresses on the ability of students to work efficiently in pairs and to have effective communication skills. |
| 5. | Standards: The course would emphasize on applying relevant standards for writing computer programs. |

Learning Outcome (LO):

| | |
|----|---|
| a. | On the successful completion of this course, students should be able to analyze computing problems for a given domain. |
| b. | The students should be able to devise algorithmic solution to solving problems in a particular domain. |
| c. | On the course completion, students should have ability to implement algorithmic solutions using a programming language. |
| d. | The students should be able to apply standards for writing programs. |
| e. | The students should have ability to collaborate and communicate efficiently in groups. |

| No. | Course Learning Outcome (CLO) Statements | Taxonomy Level | PLO |
|-----|--|----------------|-----|
| 1 | Apply the basic concepts of structured programming to write C++ programs. | C3 | 5 |
| 2 | Write C++ programs that employ the use of pointers, and structures (records). | C3 | 5 |
| 3 | Apply the basic concepts of Object Oriented Programming (class, object, attributes, data hiding, constructors, destructor, static, constant, object as argument, array of objects) to write C++ programs by using single/multiple objects. | C3 | 5 |
| 4 | Use the concepts of operator overloading, inheritance, aggregation, friend function, virtual function and polymorphism to write C++ programs. | C3 | 5 |
| 5 | Demonstrate the use of C++ generic programming concepts, function and class templates. | C3 | 5 |

Courseware Structure: (Mark X where applies)

| Lecture (Lect) | Multimedia (MM) | Exercise (s) (Exer) | Labs (Lab) | Case Studies (CAS) | Assignment (s) (Assign) | Group Tasks | Any other Medium |
|-------------------|--------------------|---------------------------|---------------|--------------------------|-------------------------------|-------------|------------------|
| X | X | X | X | | X | X | |

| Weeks | Contents/Topics | Course Activity |
|---------|--|-----------------|
| Week-01 | <ul style="list-style-type: none"> Course Introduction Revision of Basic C++ Concepts | |
| Week-02 | <ul style="list-style-type: none"> Pointers In C++ Pointer Variable Declarations and Initialization Referencing/Dereferencing, Pointer Arithmetic Pointers & Functions | |



| | | |
|----------------|--|--|
| Week-03 | <ul style="list-style-type: none">Dynamic Memory AllocationDynamic VariablesDynamic Multi-dimensional Arrays | |
| Week-04 | <ul style="list-style-type: none">Structures in C++ LanguageMember Variables & Member FunctionsArrays vs. Structures and Arrays of StructuresStructs and Pointer Variables | |
| Week-05 | <ul style="list-style-type: none">Object Oriented Programming (OOP) & Procedural ProgrammingObject-Oriented Design (OOD) and OOPIntro to Classes & ObjectsMember Functions: Access Functions (Accessors and Mutators) Utility Functions | |
| Week-6 | Mid Exam -I | |
| Week-07 | <ul style="list-style-type: none">Static members and functionsConstant members and this pointer | |
| Week-08 | <ul style="list-style-type: none">Constructor, Destructor | |

| | | |
|-----------------|---|--|
| | <ul style="list-style-type: none"> Classes and Pointer Variables Copy Constructor, Overloading Constructors Shallow Copy & Deep Copy (w.r.t. Objects) Inheritance Accessing public, private and protected base class members | |
| Week-09 | <ul style="list-style-type: none"> Inheritance with default/non-default constructors Destructors with inheritance Destructors for the dynamically allocated objects Function Overriding/Redefining Dynamic allocation in base and derived classes Inheritance – Multiple inheritance – Ambiguity errors with detailed examples. Types of inheritance (Public, Private & Protected) | |
| Week-10 | <ul style="list-style-type: none"> Composition: Association & Aggregation Friend Functions and classes | |
| Week-11 | Mid Exam –II | |
| Week -12 | <ul style="list-style-type: none"> Operator overloading – overview Operator overloading and Friend functions. | |
| Week-13 | <ul style="list-style-type: none"> Operator overloading - overloading basic operators with detailed examples. | |
| Week 14 | <ul style="list-style-type: none"> Polymorphism – Introduction (Virtual functions) Polymorphism Abstract and concrete classes | |
| Week-15 | <ul style="list-style-type: none"> Abstract Classes & pure Virtual Functions (Interface vs. Implementation) | |
| Week-16 | <ul style="list-style-type: none"> C++ Templates – Introduction and usage with detailed examples | |
| Week-17 | Final Exam | |

| Grading Criteria | | |
|------------------|---|------------------|
| Absolute | X | RELATIVE Grading |

Tentative Marks Distribution:

| Particulars | % Marks | *Weight Ranges |
|--|------------|----------------|
| 1. Assignments | 10 | 10 ~ 20 |
| 2. Quizzes (Not decided, Upon Situation) | 15 | 10 ~ 20 |
| 3. Mid Term 1 | 12.5 | 10 ~ 15 |
| 4. Mid Term 2 | 12.5 | 10 ~ 15 |
| 5. Project | 10 | 10~20 |
| 6. Final Exam | 40 | 40 ~ 60 |
| Total:- | 100 | 100 |

Planned Courseware Events:

| Particulars | Planned Items | Remarks |
|----------------|---------------|------------------------|
| 1. Quizzes | >= 5 | Unannounced quizzes |
| 2. Assignments | >= 5 | Individual assignments |

| | |
|--------------------------------------|--|
| QUALIFYING ATTENDANCE | <p>You must attend every class for your own personal benefit. Please refer to university policy of minimum attendance requirement.</p> <p>Failing to confirm qualifying attendance threshold, the student will stand debarred from sitting in the examination and assigned with “F” Grade.</p> |
| Academic and Moral Integrity: | <ol style="list-style-type: none"> All assignments should be your own work (or your group’s when approved). PLAGIARISM will be awarded with “F” grade and/or reported to the University for academic and moral misconduct. Missed quizzes/assignments will not be rescheduled. Copied assignments shall not be accepted and will result in deduction of marks already scored. |

Instructions / Suggestions for STUDENTS for satisfactory progress in this course:

- ✓ On average, most students find at least three hours outside of class for each class hour necessary for satisfactory learning.
- ✓ The homework assigned is a minimum. You should always work extra hours on your own.
- ✓ Use the few minutes you usually have before the start of each class to review the prior meetings' notes and homework. This will save us valuable in-class time to work on new material.
- ✓ Develop a learning habit rather than memorizing; work in groups, whenever appropriate.
- ✓ Apply the learned principles and gained knowledge; be creative in thinking.
- ✓ **Assignments/ Activities:** They are not meant simply for grades, but to reinforce your learning. Assignments are due on time. Each day late will lower your assignment grade by 30%. You can submit assignment till three days later after submission date.