



### CS1004 - Object Oriented Programming

Form number	COURSEWARE OUTLINE/ DOCUMENT (Tentative)				
COURSE INSTRUCTOR INFORMATION	Name	<b>Engr. Muhammad Usman Malik</b>			
	email ID	<a href="mailto:usman.malik@nu.edu.pk">usman.malik@nu.edu.pk</a>			
	Contact	-----		Computer Engineering Dept.	
DEGREE INFORMATION	Program	Batch	Section(s)	Semester	Spring
	BS(CS)	2024	BSCS-2A	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 Materials	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"> <li>Course Introduction</li> <li>Revision of Basic C++ Concepts</li> </ul>	
Week-02	<ul style="list-style-type: none"> <li>Pointers In C++</li> <li>Pointer Variable Declarations and Initialization</li> <li>Referencing/Dereferencing, Pointer Arithmetic</li> <li>Pointers &amp; Functions</li> </ul>	



<b>Week-03</b>	<ul style="list-style-type: none"><li>• Dynamic Memory Allocation</li><li>• Dynamic Variables</li><li>• Dynamic Multi-dimensional Arrays</li></ul>	
<b>Week-04</b>	<ul style="list-style-type: none"><li>• Structures in C++ Language</li><li>• Member Variables &amp; Member Functions</li><li>• Arrays vs. Structures and Arrays of Structures</li><li>• Structs and Pointer Variables</li></ul>	
<b>Week-05</b>	<ul style="list-style-type: none"><li>• Object Oriented Programming (OOP) &amp; Procedural Programming</li><li>• Object-Oriented Design (OOD) and OOP</li><li>• Intro to Classes &amp; Objects</li><li>• Member Functions: Access Functions (Accessors and Mutators) Utility Functions</li></ul>	
<b>Week-6</b>	<b>Mid Exam -I</b>	
<b>Week-07</b>	<ul style="list-style-type: none"><li>• Static members and functions</li><li>• Constant members and this pointer</li></ul>	
<b>Week-08</b>	<ul style="list-style-type: none"><li>• Constructor, Destructor</li></ul>	

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



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
<b>Total:-</b>	<b>100</b>	<b>100</b>

**Planned Courseware Events:**

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

**QUALIFYING ATTENDANCE**

You must attend every class for your own personal benefit. Please refer to university policy of minimum attendance requirement.  
Failing to confirm qualifying attendance threshold, the student will stand debarred from sitting in the examination and assigned with "F" Grade.

**Academic and Moral Integrity:**

1. 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.
2. Missed quizzes/assignments will not be rescheduled.
3. 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.