



COURSE DESCRIPTION FORM

INSTITUTION National University of Computer and Emerging Sciences (NUCES-FAST)
BS(CS), BS(CY), BS(SE), BS(AI)

**PROGRAM (S) TO BE
EVALUATED**

A. Course Description

Course Code	CS-1004
Course Title	Object-oriented Programming
Credit Hours	3+1
Prerequisites by Course(s) and Topics	Programming Fundamentals (CS-1002)
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	<p>Theory: Mid-1: 15 Mid-2: 15 Quizzes: 12 (4 total) Assignments: 8 (2 total : 4+4) Final: 50</p> <p>Lab: Lab Activities: 20 (2 each and best 10) Midterm : 20 Project : 10 Final : 50</p>



National Computing Education Accreditation Council

NCEAC



Course Coordinator	Sumaiyah Zahid						
URL (if any)	-						
Current Catalog Description	-						
Textbook (or Laboratory Manual for Laboratory Courses)	<p>Textbook:</p> <p>1. "Problem Solving with C++", 9e Global Edition, Walter Savitch, ISBN-13:9781292018249, Addison-Wesley, 2015.</p> <p>2. C++ How to program By Deitel & Deitel.</p> <p>Reference books:</p> <p>1. The C++ Programming Language by Bjarne Stroustrup.</p> <p>2. Object Oriented Software Engineering by Jacobson.</p> <p>3. C# 4.0: The Complete Reference by Herbert Schildt</p>						
Reference Material	GCR						
Course Goals	<p>A. Course Learning Outcomes (CLOs) with Bloom's Taxonomy Levels</p> <ol style="list-style-type: none">1. Discuss knowledge of underlying concepts of object-oriented paradigm like abstraction, encapsulation, polymorphism, inheritance etc. (C-2)2. Identify real world problems in terms of objects rather than procedure. (C-4)3. Illustrate Object-Oriented design artifacts and their mapping to Object-Oriented Programming using C++. (C-3)4. Design and assess small and medium scale C++ / C# programs using object-oriented programming principles. (C-6)5. Synthesize programs using Generic Programming and exception handling. (C-6) <p>B. Program Learning Outcomes</p> <table border="1"><tr><td>1. Computing Knowledge</td><td>Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solution of complex computing problems.</td><td>✓</td></tr><tr><td></td><td></td><td>✓</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.	✓			✓
1. Computing Knowledge	Apply knowledge of mathematics, natural sciences, computing fundamentals, and a computing specialization to the solution of complex computing problems.	✓					
		✓					



National Computing Education Accreditation Council

NCEAC



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 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	



National Computing Education Accreditation Council

NCEAC



HEC FORM 001.D

	9. Individual and Team Work	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 or a team.											
	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. Relation between CLOs and PLOs (CLO: Course Learning Outcome, PLOs: Program Learning Outcomes)													
		PLOs											
		1	2	3	4	5	6	7	8	9	10	11	12
C L O s	1	✓											
	2	✓											
	3		✓										
	4			✓									
	5		✓										

Topics Covered in the Course, with Number of Lectures

Week	Topic	CLO	Lab Topic	Assessment
------	-------	-----	-----------	------------



National Computing Education Accreditation Council

NCEAC



2021-2022 Academic Year

on Each Topic (assume 15-week instruction and one-hour lectures)	1	Introduction to OO paradigm	1	Introduction to IDE, skeleton of C++ program, double pointers, 2d arrays, basic I/O in C++ C++ data types, functions, struct revisited based on real world use cases	Assignment 1 Quiz 1 Week 3	
	1	Comparison from sequential & procedural paradigms	1			
	1	Data Abstraction Encapsulation	1			
	2	Introduction to Objects in real world	1,2			
		Introduction to classes and objects	1,2,3			
		Access Control	1,2,3			
		Constructors & its types, Destructor	1,3,4			
	3	Setters & Getters	1,3,4	Working with classes and Constructors, setters and getters, Working with access modifiers		
		Member initialization list	1,3			
		Constants, Constants with pointers, constant functions	1,3			
	4	Static data and member functions,	1,3	Static and constant keywords, This pointer		
		Inline functions, This pointer	1,3			
	5	Array of objects Has-a relation	1,2,3,4	Array of objects Has-a relation		
	6	Mid I Exam				
	7	Introduction of Inheritance	1,2,3,4	Member initialization list Inheritance	Assignment 2 Quiz 2 Week 7 Quiz 3 Week 10	
		Types of inheritance				
	8	Data and code hiding	1,2,3,4	Polymorphism, Function overloading and overriding		
		Polymorphism in OOP	1,2,3,4			
		Function overriding and overloading	1,2,3,4			
	9	Friend function	1,2,3,4	Lab Mid		
		Operator overloading	1,2,3,4			
	10	Multiple inheritance & its issues (Diamond Problem)	1,2,3,4	Friend classes, Friend functions, operator overloading		
		Virtual inheritance	1,2,3,4			
		Virtual functions	1,2,3,4			
	11	Abstract classes & Interfaces	1,2,3,4	Abstract Classes and virtual functions		
	12	Mid II Exam				
	13		1,2,3,4	Filing and I/O stream		



National Computing Education Accreditation Council
NCEAC



HEC Approved

	14	Introduction to filing Generics	5	Working with template functions and template classes	Quiz 4	
		Generics and Templates	5			
	15	Introduction to exception handling	1,2	Project Submission & Project demo		
		STL (Vector, List)	1,2			
	16	Final lab exam				
		Final Exam				
Laboratory Projects/Experiments Done in the Course	1					
Programming Assignments Done in the Course	3 Assignments					
Class Time Spent on (in credit hours)	Theory	Problem Analysis	Solution Design	Social and Ethical Issues		
	15	15	13	0		
Oral and Written Communications	Every student is required to submit at least <u>1</u> written report of typically <u>2</u> pages and to make <u>1</u> oral presentations of typically <u>10</u> minute's duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.					

Instructor Name: Sumaiyah Zahid

Date: 15th January, 2026