National University of Computer and Emerging Sciences – NUCES - Karachi

COURSE DESCRIPTION FORM

Departments: BS(CS), BS(SE), BS(AI), BS(DS), BS(CY)

PROGRAM(S) TO BE EVALUATED

Lab Description

<table>
<tr><td>Course Code</td><td>CL-1004</td></tr>
<tr><td>Course Title</td><td>Object-Oriented Programming</td></tr>
<tr><td>Credit Hours</td><td>3 + 1</td></tr>
<tr><td>Prerequisites by Course(s) and Topics</td><td>Programming Fundamentals (CL-1002)</td></tr>
<tr><td>Assessment Instruments with Weights</td><td>
<table>
<tr><th>Assessment</th><th>Weightage</th></tr>
<tr><td>Lab Activities</td><td>20 (12 Labs, 1.66 each)</td></tr>
<tr><td>Lab Mid-Exam</td><td>20</td></tr>
<tr><td>Project</td><td>10</td></tr>
<tr><td>Lab Final-Exam</td><td>50</td></tr>
</table>
</td></tr>
<tr><td>Textbook (or Laboratory Manual for Laboratory Courses)</td><td>Textbooks:
1. “Problem Solving with C++”, 9e Global Edition, Walter Savitch, ISBN- 13:9781292018249, Addison-Wesley, 2015.
2. “C++ How to program” by Deitel & Deitel.
Reference books:
1. “The C++ Programming Language” by Bjarne Stroustrup.
2. “Object Oriented Software Engineering” by Jacobson.
3. “C# 4.0: The Complete Reference” by Herbert Schildt.</td></tr>
<tr><td>Course Goals</td><td>
<table>
<tr><th>A. Lab Learning Outcomes (LLOs) with Bloom's Taxonomy Levels</th></tr>
<tr><td>1. Discuss knowledge of underlying concepts of object-oriented paradigm like abstraction, encapsulation, polymorphism, inheritance etc. (L-2)
2. Identify real world problems in terms of objects rather than procedure. (L-4)
3. Illustrate Object-Oriented design artifacts and their mapping to Object-Oriented Programming using C++. (L-3)</td></tr>
</table>
</td></tr>
</table>

4. **Design** and assess small and medium scale C++ / C# programs using object-oriented programming principles. **(L-6)**
5. **Synthesize** programs using Generic Programming and exception handling. **(L-6)**

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 analyze complex computing problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, and computing sciences.	✓
3. Design/Develop Solution	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 & 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 & 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 & 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 LLOs and PLOs** (LLO: Lab Learning Outcomes, PLOs: Program Learning Outcomes)

		PLOs											
		1	2	3	4	5	6	7	8	9	10	11	12
LLOs	1	✓											
	2	✓											
	3		✓										
	4			✓									
	5		✓										

Topics Covered in the Course with number of Labs on each Topic

Week	Lab	LLO
01	Introduction to C++ Programming: Program Structure, Basic Input/Output, Data Types, Functions, and Arrays.	1
02	Exploring Pointers, Dynamic Memory Allocation, and Structs in C++.	1
03	Fundamentals of Object-Oriented Programming: Classes, Objects, Access Modifiers, and Header Files.	2
04	Dive into Constructors, Destructors, and the "this" Keyword in C++.	2

	05	Exploring Advanced Access Modifiers, Static Members, Object Relationships, and Arrays of Objects.	2
	06	**Theory Mid-Exam 01**	
	07	Static and Constant Functions with Member Initialization Lists in C++.	2
	08	Inheritance, Inheritance Types and Resolving the Diamond Problem.	3
	09	**Lab Mid-Exam**	
	10	Polymorphism, Virtual Functions, Function Overloading and Overriding, Operator Overloading, and Binding in C++.	3
	11	Friend Functions, Friend Classes, Virtual Inheritance.	3
	12	**Theory Mid-Exam 02**	
	13	Abstract Classes and Pure Virtual Functions for Advanced Object-Oriented Design.	3
	14	Templates and Exception Handling.	4
	15	Practical File Handling and IOStream Techniques in C++.	4
	16	**Lab Final-Exam**	

Instructor Name: **Talha Shahid**

Instructor Signature: _____

Dated: **20th Jan, 2025**