SALIM HABIB UNIVERSITY

COURSE OUTLINE

Institution: Salim Habib University

Dept-Faculty: Department of Computer Science, Faculty of Information Technology

Course Name: Object Oriented Programming (3+1)

Course Code: CSC- 103 Session: Fall 2025

Instructor Name: Ateeque Rahman

Program to be evaluated: BS (SE)

A. Course Objectives

The main objective of this course is to introduce the organization and operation of computer systems at the assembly language level. Covers mapping of statements and constructs in a high-level language onto sequences of machine instructions, as well as the internal representation of simple data types and structures. Offers programming practice with an assembly language to provide practical application of concepts presented in class.

Course Learning Outcomes (CLOs):

Upon successful completion of this course, a student will be able to:

Sr. No	CLOs	Domain	Taxonomy Level	GAs Mapping
01	**Understand** principle of Object– Oriented	**Cognitive**	2	2
02	**Identify** the object & their relationships to build object-oriented solution.		3	3
03	**Model** a solution for a given problem using object-oriented principles		5	4
04	**Examine** an object – oriented solution		4	3

B. Course Description

Course Code	CSC-103
Course Title	Object Oriented Programming
Credit Hours	3+1
Contact Hours	16x3=48
Prerequisites by Course(s) and Topics	NIL

Course Catalog Number	--	
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	**Mid-term paper**	20
	Final paper	40
	Sessional	20
	Lab	20
	Total	100%
Textbook (or - **Laboratory Manual** for Laboratory Courses)	1. Beginning C++ 2, 7th Edition by Ivor Horton 2. An Introduction to Object Oriented Programming with C++, 5th Edition by C. Thomas Wu	
Reference Material	1. Effective C++" by Joshua Bloch 2. C++ Generics and Collections" by Maurice Naftalin and Philip Wadler	
Course Goals	An Object-Oriented Programming (OOP) course aims to provide students with a comprehensive understanding of OOP concepts, mastery of object-oriented design principles, proficiency in a programming language, problem-solving skills, the ability to develop modular and reusable code, experience in collaborative programming, debugging and troubleshooting proficiency, knowledge of best practices and software engineering principles, exposure to object-oriented analysis and modeling, and preparation for software development careers. Through a combination of theoretical concepts, practical programming assignments, and collaborative projects, the course equips students with the necessary skills and knowledge to design and develop robust, maintainable, and scalable software systems using object-oriented programming techniques.	
Topics Covered in the Course, with Number of Lectures on Each Topic (assume 15-week instruction and one-hour lectures)	• Basics of C++ • Introduction to object-oriented design • History and advantages of object-oriented design • Introduction to object-oriented programming concepts • Classes, objects, data encapsulation, constructors, destructors, access modifiers, const vs non-const functions, static data members & functions • Function overloading and Operator overloading • Identification of classes and their relationships • Composition, aggregation, inheritance, multiple inheritance, polymorphism, abstractclasses and interfaces • Generic programming concepts, • Function & class templates • Standard template library	

	• Object streams, data and object serialization using object streams • Exception handling
Laboratory Projects/Experiments Done in the Course	C++ based Projects
Programming Assignments Done in the Course	Assignment and Quizzes

Class Time Spent on (in hours)	**Theory**	**Problem Analysis**	**Solution Design**	**Quizzes**	**Revision**
	20	10	10	5	3

Oral and Written Communications	In whole semester, every student was required to submit at least _01_ written reports typically of _10_ pages and to make _01_ oral presentation of typically _10_ minutes duration. Every student is required to participate in _01_ special Session, Class Discussion – Flip Style classroom. For the official communication (lectures, assignments, solving students' queries, LMS portal is used.)

C. Course Outline (Week-wise)

Week	Duration	Topics Covered	Evaluation and Activity	Lecture Method and Technology Used	Related Reading Materials
1	3 hours	Introduction of C++ History of C++ Need of C++ Bytecode Compilers, Interpreters & Cross Compilers Multithreaded Application Security Portability C++ vs. C++	- Q&A session - Reading assignment - Group discussion - Code walkthrough - Exercises (Hands-on)	- Lecture presentation - Slides - Whiteboard	- "C++: A Beginner's Guide" by Herbert Schildt
2-3	3 hours	C++ Building Blocks Built-in Data Types Literals	- Exercises (Hands-on)	- Lecture presentation - Slides	- "C++: The Complete

		Naming Conventions Control Structures	- Code walkthrough - Q&A session - Group discussion	- Whiteboard	Reference" by Herbert Schildt
4	3 hours	Arrays & Strings One-dimensional & Multidimensional Arrays Array list, Loop (for each loop) String Methods Introducing Methods	- Sorting and Searching exercises (Hands-on) - Code walkthrough - Q&A session - Group discussion - Assignment	- Lecture presentation - Slides - Whiteboard	- "Effective C++" by Joshua Bloch
5	3 hours	Introducing OOP Class & Object Constructor & Finalizer	- Exercises (Hands-on) - Code walkthrough	- Lecture presentation - Slides	- "Head First C++" by Kathy Sierra and Bert Bates
6	3 hours	Destructors and Encapsulation Types of Class Members Operator Overloading Friends Class Global Variables and Functions Pointers	- (Hands-on) - Quiz - Exercise - Code walkthrough - Q&A session - Group discussion	- Lecture presentation - Slides - Whiteboard	- "C++ in a Nutshell" by Benjamin J. Evans
7	3 hours	Inheritance Superclass and Base Class Sequence of Constructors Call in Hierarchy Types of Inheritance UML diagrams	- Final Keyword exercises (Hands-on) - Code walkthrough - Q&A session - Group discussion - Quiz	- Lecture presentation - Slides - Whiteboard	- "Thinking in C++" by Bruce Eckel
8		Mid Exams			
9	3 hours	Packages Controlling Visibility and Name Conflicts using Packages	- Creating and Using Packages exercises (Hands-on) - Code walkthrough	- Lecture presentation - Slides	- "C++ Package" by Herbert Schildt

10	3 hours	Casting, Binding, Coupling and their Types	- Q&A session	- Lecture presentation	- "C++ Generics and Collections" by Maurice Naftalin and Philip Wadler
11-12	3 hours	Polymorphism and its Types	- Overriding and Overloading exercises (Hands-on) - Runtime and Compile Time exercises (Hands-on) - Kinds of Polymorphism exercises (Hands-on)	- Lecture presentation - Slides - Whiteboard	- "Effective C++" by Joshua Bloch
13	3 hours	Abstraction Interface and Abstract Class Concrete Methods vs. Abstract Methods	- Exercises (Hands-on) - Quiz - Exercises (Hands-on)	- Lecture presentation - Slides	- "C++: A Beginner's Guide" by Herbert Schildt
14	3 hours	Exception Handling	- Try/Catch Blocks exercises (Hands-on) - Throwing and Catching an Exception exercises (Hands-on) - Creating Your Own Exception Class exercises (Hands-on)	- Lecture presentation - Slides - Whiteboard	- "Effective C++" by Joshua Bloch
15	3 hours	File I/O and Storage	- File Handling exercises (Hands-on) - Quiz	- Lecture presentation	- "C++ I/O" by Elliotte Rusty Harold
16		Final Exams			

D. Lab Outline (Week-wise)

Lab	Duration	Topics Covered	Evaluation and Activity/Tools
1	3 hours	Fundamental of C++	- Q&A session - Reading assignment - Group discussion - Code walkthrough - Exercises (Hands-on)
2	3 hours	Classes and Objects	- Q&A session - Reading assignment - Group discussion - Code walkthrough - Exercises (Hands-on)
3	3 hours	Constructors	- Q&A session - Reading assignment - Group discussion - Code walkthrough - Exercises (Hands-on)
4	3 hours	Encapsulation	- Code walkthrough - Exercises (Hands-on)
5	3 hours	Inheritance	- Code walkthrough - Exercises (Hands-on)
6	3 hours	Overloading & Overriding	- Code walkthrough - Exercises (Hands-on)
7	3 hours	Packages	- Code walkthrough - Exercises (Hands-on)
8	3 hours	Abstract Classes and Methods	- Code walkthrough - Exercises (Hands-on)
9	3 hours	Interfaces	- Code walkthrough - Exercises (Hands-on)
10	3 hours	Exception Handling	- Code walkthrough - Exercises (Hands-on)
11	3 hours	Filing	- Code walkthrough - Exercises (Hands-on)
12	3 hours	GUI	C++FX / C++
13	3 hours	Version Control	Github
14	3 hours	Agile Management	JIRA
15	3 hours	Coding Platforms	Replit

1. Class Participation

The instructor highly recommends a regular class attendance.

2. Late Policy

All assignments must be submitted electronically before deadline. *"Internet / MS Teams not working"* is not an acceptable excuse for delay in submission. There is ***NO*** late submission policy.

3. Grading

During the semester the student can only earn *points* and *not* grades. At the end of the semester, the weighted sum of all points is mapped to a grade. The weighting is as follows:

Sr. No	Item	Weightage
1	**Mid-term paper**	20
2	**Final paper**	40
3	**Sessional**	20
4	**Lab**	20

ASSESSMENT-BASED CLO, BL, GAS MAPPING:

Assessment Instruments	CLO Covered	BL Covered	GAs Covered
Assignment	2,3	3,5	3,4
Quiz	1,2	2,3	2,3
Project	1,3,4	2,5,6	2,4,5
Mid Exam	2,3	3,5	3,4
Lab/Project	1,3	2,5	2,4
Final Exam	2,4	3,6	3,4