

Course Outline

Object Oriented Programming				
Course Information	Course Title			
	Course ID	CS-2104/ CS-1154	Course Type	Computing Core
	Credit hours	4 (3-3)	Hours per week	3-3
	Programs	ADP Computing (CS, AI, DS) ADP Computer System	Preferred Semester	2 nd
Course Description	<p>This course is designed as an entry level programming course for students who have prior programming experience. This course introduces the concepts of object-oriented programming to students with a background in the procedural paradigm.</p> <p>The course begins with a brief review structured data type. It then moves on to introduce the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Other topics include an overview of programming language principles, simple analysis of algorithms, and an introduction to software engineering issues.</p> <p>Brief review of control structures, functions, and primitive data types</p> <p>Object-oriented programming: Object-oriented design; encapsulation and information-hiding; separation of behavior and implementation; classes, subclasses, and inheritance; polymorphism; class hierarchies; dynamic memory allocation, File handling</p>			
Course Objectives	The objective of this course is to enable students;			
	No.	Objective		
	1.	At the end of the class, we expect people to have a good understanding about the concept of object-oriented programming using C++, be able to write and read basic C++ code.	PLO 1,	
	2.	Using inheritance and polymorphism to manage the source code.	PLO 4, 5	
Course Learning Outcomes (CLO)	3.	Student will be motivated and encouraged to choose a practical project which should be management system or transactional system.	PLO 1-6	
	At the end of this course students will be able to demonstrate;			
	No.	Outcome		Relation with SLO/PLO
	1.	Understand the relative merits of C++ as an object oriented		PLO 1,

	programming language					
2.	Understand the features of C++ supporting object-oriented program and produce object-oriented software using C++	PLO 4, 5, 6, 10, 11				
3.	Understand how to apply the major object-oriented concepts to implement object-oriented programs in C++, encapsulation, inheritance and polymorphism	PLO 8, 12				
4.	Understand advanced features of C++ specifically stream I/O, templates and operator overloading	PLO 6, 7, 9				
Lecture type	Class room Lectures, Lab Sessions, Project Presentation					
Prerequisites	Programming Fundamental I					
Follow up Courses	Data Structures					
Course Software or Tool	Visual Studio, Dev C++					
Textbook	Title	Edition	Authors	Publisher	Year	ISBN
	Object Oriented Programming	4 th	Robert Lafore	Sams	December 2001	0-672-32308-7
References	C++ Programming: Program Design Including Data Structures	6 th	D.S. Malik			978-1-13362628-1
Assessment Criteria (100%)	Assessment	Weight		Used to attain CLO		
	Assignments & Quiz	10%		CLO1-CLO4		
	Project & Presentation	10%		CLO1-CLO4		
	Lab	15%		CLO1-CLO4		
	Mid Term	25%		CLO1-CLO4		
	Final	40%		CLO1-CLO4		
Methods of Evaluation	Assignments, Quizzes, Project, Midterm paper, Final term paper					

Week No.	Topic	Lecture No.	Lecture Contents	Relation with CLO	Task
W1.	Introduction to OOP	L1.	<ul style="list-style-type: none"> Module Discussion Introduction to OOP Primitive Data types and User defined datatypes Structure Declaration of a simple structure Defining a structure variable Accessing members of the structure Initialization of a structure variables 	CLO 1	C++ Object Basics: Functions, Recursion, and Objects Read: Week 1 Notes
		L2.	<ul style="list-style-type: none"> Nested Structure Declaration, Definition, Accessing members and initialization of nested structure Enumerations 		
W2.	Objects and Classes	L3.	<ul style="list-style-type: none"> Characteristics of Object-Oriented Languages Defining the Class Using the Class Calling Member Functions 	CLO 2	Read: Week 3 Notes
		L4.	<ul style="list-style-type: none"> Examples of class Member Functions Defined Outside the Class 		Assignment: C++ Classes and Objects Coursera Project
W3.	Constructors	L5.	<ul style="list-style-type: none"> Objects as Function Arguments Objects as Arguments 	CLO 2	Read: Week 4 Notes
		L6.	<ul style="list-style-type: none"> Defining Constructor Overloaded Constructors The Default Copy Constructor 		
W4.	Static Class Data	L7.	<ul style="list-style-type: none"> Defining Destructors Example of Destructors 	CLO 2	
		L8.	<ul style="list-style-type: none"> Uses of Static Class Data An Example of Static Class Data 		
W5.	Inheritance	L9.	<ul style="list-style-type: none"> Introduction to inheritance Derived Class and Base Class 	CLO 2, 3	
		L10.	<ul style="list-style-type: none"> Accessing Base Class Members The protected Access Specifier 		Object-Oriented C++: Inheritance and Encapsulation Read: Week 1 notes
W6.	Type of Inheritance	L11.	<ul style="list-style-type: none"> Derived Class Constructors Base Class Constructors 	CLO 2, 3	
		L12.	<ul style="list-style-type: none"> Public Inheritance Protected Inheritance Private Inheritance 		Read: Week 2 Notes 1, 2
W7.		L13.	<ul style="list-style-type: none"> Levels of Inheritance 	CLO 2, 3	

	Levels of Inheritance	L14.	<ul style="list-style-type: none"> • Multiple Inheritance • Member Functions in Multiple Inheritance 	CLO 2, 3	Read: Week 2 Notes 3
W8.	Polymorphism	L15.	<ul style="list-style-type: none"> • Polymorphism • Advantages of Using Polymorphism 		
		L16.	<ul style="list-style-type: none"> • Pointers • Pointer to Objects 		
W9.	Mid Term Exam Week	L17.	Mid Term Examination		
		L18.			
W10.	Polymorphism	L19.	<ul style="list-style-type: none"> • Pointer to Objects with coding example • Early binding concept 	CLO 3	Read: Week 3 Notes
		L20.	<ul style="list-style-type: none"> • Polymorphism with Virtual Function Concept • Implementation 	CLO 3	
W11.	Virtual Functions	L21.	<ul style="list-style-type: none"> • Virtual Function • Abstract Base Class & Concrete Derived Class 	CLO 3	Assignment: C++ Inheritance, Aggregation and Composition
		L22.	<ul style="list-style-type: none"> • Normal Member Functions Accessed with Pointers • Virtual Member Functions Accessed with Pointers 	CLO 3	
W12.	Late Binding & Friend Functions	L23.	<ul style="list-style-type: none"> • Abstract Classes and Pure Virtual Functions • Virtual Destructors 	CLO 3	
		L24.	<ul style="list-style-type: none"> • Virtual Base Classes • Friend Function • Friend Class 	CLO 2, 3	
W13.	Static Functions	L25.	<ul style="list-style-type: none"> • Static Functions • Accessing static Functions 	CLO 2, 3	
	Unary Operators	L26.	<ul style="list-style-type: none"> • The operator Keyword • Overloading Unary Operators 	CLO 2, 3	
W14.	Overloading Binary Operators & File Handling	L27.	<ul style="list-style-type: none"> • Operator Arguments • Operator Return Values 	CLO 4	
		L28.	<ul style="list-style-type: none"> • Arithmetic Operators, Concatenating Strings • Multiple Overloading • Comparison Operators, 	CLO 4	
W15.	Overloading Binary Operators & File Handling	L29.	<ul style="list-style-type: none"> • Arithmetic Assignment Operators • File Handling 	CLO 4	
		L30.	<ul style="list-style-type: none"> • Writing Data, Reading Data • Appending files, Deleting Records 	CLO 4	
W16.	Function Templates	L31.	<ul style="list-style-type: none"> • Function Templates, A Simple Function Template, Function Template Syntax, What the Compiler Does, The Deciding Argument. Template Arguments Must Match, Why Not Macros? Class Templates. 	CLO 4	
		L32.	<ul style="list-style-type: none"> • Exception Handling. Try block, catch block and throw statement. Multiple Exception Handling 		
W17	Project Presentation	L33.	<ul style="list-style-type: none"> • Project Presentation 	CLO 1-4	
		L34.	<ul style="list-style-type: none"> • Project Presentation 		
W18.	Final Exam				

Object Oriented Programming Lab

Week #	Topic	Details	Homework
1.	Introduction to the Course and C++ Programming	<ul style="list-style-type: none"> • Course Outline • Assessment Policies • History of Programming Languages • Features of C++ • Overview of basic C++ concepts • Required tools and their configurations • Edit, compile, and run C++ application 	None
2.	Structures	<ul style="list-style-type: none"> • Declaration of a simple structure • Defining a structure variable • Nested Structures 	Textbook Exercise
3.	Introduction to Object Oriented Paradigm	<ul style="list-style-type: none"> • Classes and Objects • Object Oriented vs Structured Programming • Member functions and Data members 	
4.	Object Oriented Analysis & Design	<ul style="list-style-type: none"> • OOAD • Use of UML for OOAD 	Online Exercise
5.	Classes and Objects in C++	<ul style="list-style-type: none"> • C++ Classes & Objects • Object Instantiation • Instance Variables • Class Variables • Constructors • Instance Methods • Class Methods • The this keyword • Passing and returning objects 	Textbook Exercise
6.	Classes and Objects in C++	<ul style="list-style-type: none"> • Controlling Access to Members • Set and Get Methods • static keyword usage • const keyword usage 	
7.	Object Oriented Programming with C++	<ul style="list-style-type: none"> • Abstraction • Inheritance • Method Overriding • Method Overloading • Composition • Associations • Delegation 	Textbook Exercise
8	Polymorphism using pointers	<ul style="list-style-type: none"> • Pointers • Pointer to Objects • Examples 	Textbook Exercise
9.	Mid Term Week	<ul style="list-style-type: none"> • Mid Term Examination 	
10.	Polymorphism using Abstract Classes	<ul style="list-style-type: none"> • Polymorphism • Abstract Class • Abstract class inheritance 	Textbook Exercise
11.	Methods & Arrays: A Deeper Look	<ul style="list-style-type: none"> • static Methods, static Fields • Method Signatures • Declaring Methods with Multiple Parameters • Passing Objects to Methods • Method-Call Stack and Stack Frames • Argument Promotion and Casting • Scope of Declarations • Passing Arrays to Methods • Pass-By-Value vs. Pass-By-Reference • Arrays of Objects • Multidimensional Arrays 	Textbook Exercise

Week #	Topic	Details	Homework
12.	File Processing	<ul style="list-style-type: none"> • Files and streams • Creating a sequential file • Reading and updating a sequential file • Random access files 	Textbook Exercise
13.	Exception Handling	<ul style="list-style-type: none"> • Why Exceptions • Standard Exception Handling Options • Exception Class Hierarchy • Catching an Exception: try and catch blocks • Rethrowing exception 	Textbook Exercise
14.	Exception Handling	<ul style="list-style-type: none"> • Methods Which Throw Exceptions: the throws clause • Stack unwinding • Handling new in exception handling • Use of unique pointer • Writing Custom Exceptions 	Textbook Exercise
15.	Templates and Operator Overloading	<ul style="list-style-type: none"> • Class Templates • Writing your own template classes • Operator overloading 	Textbook Exercise
16.	Function Template	<ul style="list-style-type: none"> • A Simple Function Template, Function Template Syntax, What the Compiler Does, The Deciding Argument. Template Arguments Must Match, Why Not Macros? Class Templates. 	Textbook Exercise
17.	Project Demonstrations	<ul style="list-style-type: none"> • Development Project presentations 	
18.	Final exam	<ul style="list-style-type: none"> • Final exam 	None