University of Gujrat

Faculty of CS & IT

Title	Object Oriented Programming
Code	CS – 201
Credit hours	4
Prerequisite	Programming Fundamentals
Course Description	This course provides in-depth coverage of object-oriented programming principles and techniques using C++. Topics include classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes. The course briefly covers the mapping of UML design to C++ implementation and object-oriented considerations for software design and reuse. The course has a strong practical emphasis, and students will be required to implement OO concepts in C++ during supervised laboratory sessions and in unsupervised assignment work. In general, each class will consist of a one and a half hour lecture, and a one and a half hour laboratory session, which will be held weekly.
Course goal:	This course provides in-depth coverage of OOP using the C++ programming language. At the completion of this course the student should be comfortable coding a program using C++. He or she will know the strengths and weaknesses of the language. Students will then be exposed to OO analysis and design. C++ syntax and its idioms will be covered, with particular emphasis on how to program in C++ with an OO mindset.
Course objective	• By the end of the course, students should be able to: 1. create class hierarchies using OOP design 2. understand and apply inheritance techniques to their programs 3. overload and override methods and understand the difference 4. create modular programs using accepted structured programming • Create and use UML diagrams • Understand the strengths and weaknesses of OOP programming. • Use files, both binary and text.
Evaluation System	a) The teacher is responsible for the evaluation of work of the students of his/her class and for the grades on the basis of such evaluation. b) The number and nature of tests and assignments depends on the nature of the course. However, there will be at least two tests, mid semester and final examination in addition to class work. c) Each course will follow the weight age as under: Mid term 25% Sessional work 25% Final term 50% d) To pass a course, student must obtain 'D' grade (50% marks) with at least 20% marks separately in (i) mid term + sessional work and (ii) final term. e) The final term examination will cover the entire course.

Grading System

Marks in Percentage	Letter Grade	Numeric Value of Grade	Description
85 and above	A+	4.00	Exceptional
80-84	A	3.70	Outstanding
75-79	B+	3.40	Excellent
70-74	B	3.00	Very Good
65-69	B-	2.50	Good
60-64	C+	2.00	Average
55-59	C	1.50	Satisfactory
50-54	D	1.00	Pass
49 and below	F	0.0	Fail
	W		Withdrawal
	I		Incomplete

Class Attendance	a) A minimum of 39 contact hour (80 %) are required by the students to be eligible to sit in the final examination. b) A candidate with less than 80% of the attendance shall be dropped from the course.
Contact Persons:	Mr. Zafar Mehmood Khattak Hanan Bin Liaqat Muhammad Usman

Recommended Books Reference Books & Materials:	C++ by Robert Lafore C++ How to Program by Deitel & Deitel, Let us C++ by Yashavant Kanetkar www.deitel.com

COURSE OUTLINE

Week	Lecture	Topic	Chp No. RL
1	1	**The Big Picture** • Beginning of programming • Structured programming • Why Do We Need Object-Oriented Programming? • Object oriented programming	Chp 1
	2	• Characteristics of Object-Oriented Languages o Objects o Classes o Inheritance o Reusability o Data Abstraction o Data Encapsulation o Creating new data types o Polymorphism and overloading • Software Engineering Case Study: introduction to Object Technology and the UML	
2	3	**Structure** • Surcutre basics • Sturucture within structure • Structures and classes • Enumerations, • Software Engineering Case Study: examining the ATM Requirements Document	Chp 4
	4	**Objects and classes** o Basics of class and objects with real world example o Basics of class and objects with programming example o Data member and member function o Access specifier	Chp 6
3	5	• C++ objects as data types • Constructors • Destructors	

	6	• Object as function argument
		  o Overloaded constructor
		  o Member functions defined outside the class
		  o Objects as arguments	

4	7	• The default copy constructor • Returning objects from function • Class, object and memory • Static class data	
	8	• Const and classes o Const member functions o Const objects • Software Engineering Case Study: identifying the classes in the ATM Requirements Document, • Identifying the class Attributes • Objects states and activates	
5	9	**Functions and functions overloading** • Functions • Functions Basics • Overloaded functions o Different numbers of arguments o Different kinds of arguments	Chp 5
	10	• Inline functions • Default arguments • Variables and storage classes o Automatic variable o External variables o Static variables o Storage • Const function arguments • Software Engineering Case Study: class operation in the ATM system.	
6	11	**Operator overloading** • Overloading unary operator • Overloading binary operator • Data conversion o Conversion between basic types o Conversion between objects and basic types o Conversion between objects of different classes o Conversion: when to use what.	Chp 8
	12	• pitfall of operator overloading and conversion o use similar meanings o use similar syntax o show restraint o avoid ambiguity o not all operator can be overloaded	
7	13	Implementation of the case study.	
	14	**Inheritance:** • Inheritance basics in real world and programming • Derived class and base class o public, private & protected, Abstract Classes o Specifying the derived class o Accessing base class members o The protected access specifier	Chp 9

8	15	• Derived class constructors • Overriding member functions • Class hierarchies o Abstract base class o Constructor and member functions	
	16	• Scope resolution with overridden functions • Public and private inheritance o Access combinations o Access specifiers: when to use what • Level of inheritance	
9	17	• Multiple inheritance • Ambiguity in multiple inheritance • Containership: classes within class o Composition and aggregation • Inheritance and program development •	
		Mid term	
10	18	**Pointers** • Pointer basics concepts • Addresses and pointers • The address of operator • Pointer and arrays • Pointers and functions • Pointers and ctype string • Memory management: new and delete o The new opearaotr o The delete operator o A string class using new	Chp 10
	19	• Pointer to objects • Pointers to pointers	
11	20	**Virtual functions** • Virtual functions o Normal member function accessed with pointer o Normal member function accessed without pointer o virtual member function accessed with pointer o Virtual member functions accesses without pointer o Late binding	Chp 11
	21	o Abstract classes and pure virtual functions o Virtual destructors o Virtual base classes	
12	22	• Friend functions • Friend classes • Static functions • The this pointer	
	23	**Polymorphism,** • Type of Polymorphism – o Compile time and runtime, o Function Overloading, o Operator Overloading (Unary and Binary) Polymorphism by parameter,	Revision of previous topics with respect to polymorphism
13	24	o Pointer to objects, o this pointer, o Virtual Functions, o Pure virtual functions.	

	25	**Streams and files**
		• Stream classes
		  o Advantages of streams
		  o The stream class hierarchy
		  o The ios class
		  o The isteam class
		  o The ostram class
			Chp 12

14	26	• Disk file I/O with streams • File pointers	

	27	• Error handling in file I/O • File I/O with member functions	
15	28	**Multifile programs** • Reason for multifile program • Creating a multifile program	Chp 13
	29	o Header file o Directory o Projects • Case study	
16	30	**Templates and exceptions** • Functions templates o A simple functions template o Functions templates with multiple arguments • Class templates	Chp 14
	31	• Exception o Why do we need exception o Exception syntax o A simple exception example o Multiple exceptions with arguments	
	32	Revision	