

The Islamia University of Bahawalpur
Department of Computer Science & IT

COURSE DESCRIPTION

Course Instructor	Hina Faryal			
Contact Info:	Email ID		faryal.virgo@gmail.com	
	Cell Number		03366844967	
Course Code	COSC-1102			
Course Title	Object Oriented Programming			
Credit Hours	3+1			
Assessment Instruments with Weights (homework, quizzes, midterm, final term) etc.	<ul style="list-style-type: none">• Assignments+ Quizzes = 20%• Mid Term Exam = 30%• Final Term Exam = 50%			
Text Book	<ul style="list-style-type: none">• Object Oriented Programming In C++, 4/E by Robert Lafore, Pearson Education			
Other Reference Books	<ul style="list-style-type: none">• C++: How to Programme, 5/e, Deitel and Deitel, Prentice Hall,• Computer Science: An Object Oriented Approach Using C++ by Behrouz Forouzan, Richard Gilberg, McGraw-Hill College			
Course Goals	Main Objective of this course id to demonstrate adeptness of object oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance. Also demonstrate ability to implement one or more patterns involving realization of an abstract interface and utilization of polymorphism in the solution of problems which can take advantage of dynamic dispatching.			
Topics Covered in the course, with number of lectures on each topic	W	Topics	L	Assignments
	1.	<u>THE LANGUAGE OF OBJECT-ORIENTATION</u> <ul style="list-style-type: none">• What Is Object-Oriented Programming?• C++, JAVA and Object-Oriented Programming.• What Is an Object?• What Is a Class?• OOP Features Encapsulation• Data Hiding• The Public Interface• Relationships Among Classes• Inheritance• Polymorphism	2	
	2.	<u>Introduction to Classes, Objects and Strings</u> <ul style="list-style-type: none">• Defining a Class with a Member Function• Defining a Member Function with a Parameter Data Members: Set Member Functions and get• Access Specifiers: Public, Private Protected	2	Assignment #1

		<ul style="list-style-type: none">• Member Functions• Initializing Objects with ConstructorsPlacing a Class in a Separate File for Reusability		
	3.	<u>Introduction to Classes, Objects and Strings:</u> <ul style="list-style-type: none">• Comments• Namespaces• Simple Output• Simple Input• Definitions Near to First Use• Function Prototypes• The inline Specifier	2	
	4.	<u>FUNDAMENTALS OF CLASSES</u> <ul style="list-style-type: none">• Data Types• User Defined Data Types• Using the Class Concept• Defining a Class• public and private Access Levels• The Scope Resolution Operator ::• Using Class Objects Like Built-in Types• Scope• Constructors• Member Initialization Lists• Destructors• Pointers	2	
	5.	<u>COMPOSITION OF CLASSES</u> <ul style="list-style-type: none">• Relationships• Composition of Classes• The Point Class• The Line Class• Member Initialization Lists• Application With Composition• The Copy Constructor Under Composition	2	Assignment #2
	6.	<u>OPERATOR OVERLOADING</u> <ul style="list-style-type: none">• Introduction• Rules for Operator Overloading• Rationale for Operator Overloading• Overloading Member Functions• Overloading Non-Member Functions• friend Functions• The Copy Constructor• The Assignment Operator• Overloading []• Overloading Increment and Decrement Operators	2	
	7.	<u>OPERATOR OVERLOADING</u> const Objects and Reference Overloading Unary Operators, Overloading the Unary Prefix and Postfix ++ and – Operators, Dynamic Memory Management, Operators as Member vs. Non-Member Functions Converting Between Types, explicit Constructors and Conversion Operators Overloading the Function Call Operator ().	2	

	8.	Exception Handling <ul style="list-style-type: none">• Introduction, Rethrowing an Exception,• Stack Unwinding,• When to Use Exception Handling, Constructors,• Destructors and Exception Handling, Exceptions and Inheritance, Processing new Failures,• Class unique_ptr and Dynamic Memory Allocation,• Standard Library Exception Hierarchy.		
	9.	<u>Exception Handling</u> <ul style="list-style-type: none">• Destructors and Exception Handling, Exceptions and Inheritance, Processing new Failures,• Class unique_ptr and Dynamic Memory Allocation,• Standard Library Exception Hierarchy.	2	
	10	<u>INHERITANCE</u> <ul style="list-style-type: none">• Introduction• Public Base Classes• The protected Access Level• Member Initialization Lists• What Isn't Inherited• Assignments Between Base and Derived Objects• Compile-Time vs. Run-Time Binding• virtual Functions• virtual Destructors• Pure virtual Functions	2	Assignment #3
	11.	<u>Polymorphism</u> <ul style="list-style-type: none">• Rationale• Inheritance• Basics for Polymorphism• Introduction Relationships Among Objects in an Inheritance Hierarchy• Type Fields and switch Statements, Abstract Classes and Pure virtual Functions,• Polymorphism,• Virtual Functions and Dynamic Binding	2	
	12.	<u>Why Polymorphism?</u> <ul style="list-style-type: none">• Polymorphic functions• Through inheritance• Operator overloading• Polymorphic types• Template classes• Polymorphism and Subtypes	2	
			2	

	13.	<p style="text-align: center;"><u>Structures</u></p> <p>Unions, Bit Manipulations and Enumerations:</p> <p>Structure Definitions, Initializing Structures, Accessing Structure Members,</p> <p>Using Structures with Functions, typedef, Example: High-Performance Card Shuffling and Dealing Simulation, Unions, Bitwise</p> <p>Operators, Bit Fields, Enumeration Constants</p>		
	14.	<p style="text-align: center;"><u>I/O IN C++</u></p> <ul style="list-style-type: none">• The iostream Library• Predefined Streams• Overloading operator<<• Overloading operator>>• Manipulators• Stream States• Formatted I/O• Disk Files• Reading and Writing Objects	2	Assignment #4
	15.	<p style="text-align: center;"><u>Template Functions</u></p> <ul style="list-style-type: none">• Template Classes• Multiple Inheritance• User-Defined Conversions• Data Structures• An Iterator Class• Exceptions	2	
	16.	<p style="text-align: center;"><u>The Standard Template Library</u></p> <ul style="list-style-type: none">• Design Goals• STL Components• Iterators• list• set• map• find• merge• accumulate• Function Objects• Adaptors	2	