GC Women University SIALKOT

Course Outline

Object Oriented Programming

Basic Information

Course Name	**Object Oriented Programing**
Course Code	
Program	**BS. CS/IT**
Semester	**Spring 2022**
Credit Hours	**4(3+1)**
Pre requisites (if any)	**Programing Fundamental**
Resource Person	**Dr. M. Usman Ashraf**
Contact information	**0312-7070895**
Office Hours:	**1:pm to 4:00 pm (Daily)**

Course Description:

This course provides in-depth coverage of object-oriented programming principles and techniques using Object Oriented (OO) programming language. In this course will be using C++ as a medium. Topics include classes and objects, function overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes. The course briefly covers the mapping of UML design to Object Oriented (OO) Programming language 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.

Evaluation

All the activities held during the session will be evaluated. Final grades for the course will be awarded on the basis of the following breakdown:

Assignments:	5%
Project+ Presentation:	15% + 5%
Quizzes:	5%
Mid Term Examination:	30%
Final Examination:	40%

Total: **100%**

Learning Objectives:

Sr#	Course Learning Objectives
1	Designing classes and their functionalities using OOP design
2	Understand and apply inheritance techniques to their programs
3	Overload and override methods and understand differences between them
4	Can Exploit Power of Polymorphism in Development.
5	Create and Using UML diagrams
6	Understand the strengths and weaknesses of Structural/OO programming.
7	File Handling, binary and textual

Textbooks & Supplies:

1. **Required Textbook:** C++ by Robert Lafore
1. **Recommended textbook:** An Introduction to Object-Oriented Programming with Java, C. Thomas Wu (2010). 5th Edition. McGraw-Hill. ISBN: 9780073523309
2. C++ How to program by DEITEL AND DEITEL
3. Ivor Horton's Beginning Java, 7/e, Ivor Horton.
4. All lecture slides will be made available electronically via mail (If applicable)

Supplementary Material:

URL: www.google.com

Description: Google surfing which includes additional learning activities such as self-quizzes and outlines for the student.

Classroom Behavior:

During class all cell phones must be turned off or set to "vibrate." If you are on-call for emergencies, please let me know at the beginning of the class. Students have flexibility of 5 minutes after starting the lecture.

Participant Responsibilities:

The Participant is responsible for all information presented in class (unless told otherwise) and all information in the reading assignments, whether or not covered by the instructor. In case of absence it is the participant's responsibility to get class notes, handouts, and/or directions from a classmate. Regular student attendance and participation is very important. The material covered in the classroom will be cumulative in nature, and missing classes will tend to put a student "out of sync" in ways that won't be entirely evident until an assignment or examination comes due. If you must miss a class, please let the instructor know - a make-up class can sometimes be arranged, or classroom handouts picked up.

Honesty Policy:

A Participant found in cheating on any exam/ assignment/ project will receive no credit (i.e. no grade) for that exam/ assignment/ project. Plagiarism is a serious academic offence. Students who plagiarize material for assignments will receive a mark of zero (F) on the assignment and may fail the course. Plagiarism may result in dismissal from a program of study or the college. Plagiarism involves presenting the words of someone else as you own. Plagiarism can be the deliberate use of a whole piece of another person's writing, but more frequently it occurs when students fail to acknowledge and to document sources from which they have taken material.

Topic Breakdown

.Week	Lectures	Topics	Sessional
1	1	**Course Introduction:** Course Policies/Overview/Course Contents/Course Objectives	
	2	**Introduction to Computer:** ● Beginning of programming ● Structured programming ● Why Do We Need Object-Oriented Programming? ● Object oriented programming	Chp 1
	3	**Introduction to Programming:** ▪ 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	4	**Data types:** ▪ Getting Started o Basic program structure o Integer variables o Integer constant o Character variables o Character constant o Float variables **o** Bool variable	Chp 2

		○ Escape sequences ▪ Arithmetic operation ▪ Input/output variation	
	5	**Loops and decisions** ▪ Relational operators ▪ For loop ▪ While loop ▪ Do while loop ▪ When to use which loop	Chp 3
	6	**Decisions** ▪ If statement ▪ The if else statement ▪ The else….if statement ▪ Nested if statement ▪ Switch statement **Logical operators** ▪ AND operator ▪ OR operator ▪ Not Operator **Control statement** ▪ Break ▪ Continue ▪ Go to	**Week task**
3	7	**Structure:** ● Structure basics ● Structure within structure	Chp4
	8	▪ Structures and classes Comparison ▪ Enumerations,	**Week Task**
	9	**Functions and functions overloading:** ▪ Functions ▪ Functions Basics ▪ Overloaded functions ○ Different numbers of arguments ○ Different kinds of arguments	Chp5
4	10	▪ Inline functions ▪ Default arguments ▪ Variables and storage classes	
	11	● Variables and storage classes ○ Automatic variable ○ External variables	Quiz
	12	○ Static variables ○ Storage	
5	13	**Objects and classes:**	Chp 6

		○ Basics of class and objects with real world example ○ Basics of class and objects with programming example ○ Data member and member function ○ Access specifier	
	14	▪ C++ objects as data types ▪ Constructors	
	15	▪ The default copy constructor ▪ Returning objects from function ▪ Class, object and memory Static class data	**Week Task**
6	16	▪ Const and classes ○ Const member functions ○ Const objects ▪ Identifying the class Attributes ▪ Objects states and activates	
	17	**Operator overloading** ▪ Overloading unary operator ▪ Overloading binary operator	Chp 8
	18	▪ Data conversion ○ Conversion between basic types ○ Conversion between objects and basic types ○ Conversion between objects of different classes ▪ Conversion: when to use what.	**Week Task**
7	19	▪ pitfall of operator overloading and conversion ○ use similar meanings ○ use similar syntax ○ show restraint ○ avoid ambiguity ▪ not all operator can be overloaded	
	20	**Inheritance:** ▪ Inheritance basics in real world and programming ▪ Derived class and base class ○ public, private & protected, Abstract Classes ○ Specifying the derived class ○ Accessing base class members ○ The protected access specified ○ Derived class and base class	Chp 9
	21	● Derived class constructors ● Overriding member functions ● Class hierarchies ○ Abstract base class ○ Constructor and member functions ▪ Scope resolution with overridden functions	
8	22		

	23 24	**MID TERM WEEK**	
9	25	● Public and private inheritance o Access combinations o Access specifiers: when to use what ▪ Level of inheritance	Chp 9
	26	▪ Multiple inheritance ▪ Ambiguity in multiple inheritence	
	27	▪ Containership: classes within class o Composition and aggregation ▪ Inheritance and program development	**Week Task**
10	28	**Pointers:** ▪ Pointer basics concepts ▪ Addresses and pointers	
	29	▪ Pointer basics concepts ▪ Addresses and pointers	Chap 10
	30	▪ The address of operator ▪ Pointer and arrays	
11	31	▪ Pointers and functions ▪ Pointers and ctype string	
	32	▪ Memory management: new and delete o The new operator o The delete operator o A string class using new	
	33	● Pointer to objects ● Pointers to pointers	**Week Task**
12	34	**Virtual Functions:** ▪ Virtual functions o Normal member function accessed with pointer o Normal member function accessed without pointer	Chp 11
	35	o virtual member function accessed with pointer o Virtual member functions accesses without pointer o Late binding	
	36	o Abstract classes and pure virtual functions o Virtual destructors o Virtual base classes	**Week Task**
13	37	▪ Friend functions ▪ Friend classes ▪ Static functions ▪ The this pointer	
	38	**Polymorphism:** ▪ Type of Polymorphism – ▪ Compile time and runtime	Revision of previous topics with respect to polymorphism
	39	o Function Overloading, o Operator Overloading (Unary and Binary) Polymorphism by parameter,	**Week Task**

		o Pointer to objects, o this pointer, o Virtual Functions, o Pure virtual functions	
14	40	**Streams and files** ▪ Stream classes o Advantages of streams o The stream class hierarchy o The ios class o The isteam class ▪ The ostram class	Chp 12
	41	o The stream class hierarchy o The ios class o The isteam class ▪ The ostram class	
	42	o The stream class hierarchy o The ios class o The isteam class o The ostram class	**Week Task**
15	43	**Multi file programs** ▪ Reason for multi file program ▪ Creating a multi file program o Header file o Directory o Projects ▪ Case study	Chp 13
	44	▪ Creating a multi file program o Header file o Directory o Projects	
	45	**Case Study**	
16	46	**Templates and exceptions** ● Functions templates o A simple functions template o Functions templates with multiple arguments ● Class templates ● Exception o Why do we need exception	Chp14
	47	o Exception syntax o A simple exception example **o** Multiple exceptions with arguments	**Week Task**
	48	**Revisions :** ▪ Tying-up loose ends.	