Course Outline

Course Title	Object Oriented Programming
Course Code	CMP-142
Course Webpage	✓ Yahoo Group: http://groups.yahoo.com/group/oop_pucit/ ✓ Teacher Notes: \\printsrv\\Teacher Data\\Fareed ul Hassan ✓ CMS: online.pucit.edu.pk
Course Email	fareed@pucit.edu.pk
Instructor	Fareed Ul Hassan Baig
Teacher Assistant(s) (T.A)	BCSF12M538 Hamza Asghar BITF13M050 Muhammad Umair BCSF13M008 Muhammad Ali BITF13M008 Ammar Ahmad Tahir BCSF11M075 Ahmad Majeed
Credit Hours	3 *Theory/week:* Weight 3 Cr. hrs. Lectures: 2 Duration 1.5 hrs.
Prerequisite Course	CMP-140 Programming Fundamentals CMP-141 Programming Fundamentals Lab
Prerequisite Skill/Knowledge/Understanding	○ Strong understanding of how to attack a bigger task by dividing it into smaller tasks. ○ Student should know the concept of passing/returning objects (struct) to/from functions by knowing the pitfalls that come across while doing that. ○ Understanding the real spirit/cause of data driven programming. ○ Good concepts of Type Casting. ○ Student should be familiar with debugging process.
Follow Up	CMP-210 Data Structures and Algorithms CMP-211 Data Structures and Algorithms Lab
Program Name	BS Software Engineering /Computer Science/Information Technology
Aims and Objectives	○ To equip the learner with the philosophy and necessary skills to formulate solutions of real world problems using object oriented paradigm. ○ Justify the philosophy of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism. ○ Strong concepts of object manipulation and dynamic memory allocation within classes
Syllabus	**Topics:** Object Oriented Concepts, Terminology and Features, Class/ADT/UDT, Data Abstraction and Encapsulation, Special Methods, Constructor and Destructor; Modifier const; Array and Pointer of ADT, Composition; this Pointer, friend Function and Class, Modifier static, Operator Overloading, Class Templates, Inheritance, its types, and related Terminology, Overriding: static & dynamic/Polymorphism; Stream I/O, File Processing; Exception Handling
Text Book(s)	A. H. M. Deitel "C++ How to Program", 5th Ed., Prentice Hall, 2005. ISBN

	0-13-185757-6
Reference Material	R1. Handouts. R2. Victor Shtern, "Core C++ A Software Engineering Approach", 1st Ed., Prentice Hall PTR, 2000. ISBN: 0-13-085729-7, R3. Stephen Parata, "C++ Primer Plus", 5th Ed., Sams Publishing, 2005. ISBN 0-672-32697-3 R4. Bjarne Stroustrup, "The C++ Programming Language", 3rd Ed., Addison Wesley, 1997. ISBN 0201889544
Assessment Criteria	

Sessional 25%		Mid 35%	Final 40%
Quizzes, Assignment, Tests	15	Written Exam 35	Written Exam 40
Term Paper	0		
Project	10		
Total	25	35	40
	100		

- Sessional Marks will be updated online on the Google Docs.
  https://docs.google.com/spreadsheets/d/1AgH3PdeyWBoyursJtZF5Zl6Sjs0PVhFoKqcdvI2cZVY/edit?usp=sharing

Lecture Breakdown

Week	Lecture	Topic	Source
1	1	Course Introduction Overview of Structured Programming: Focus • Coding Approach • Pointer, Arrays, Dynamic Memory Allocation	R1: Handouts
	2	C/C++ struct keyword: Data Driven Programming • Use of struct • Passing structures by reference/value	Your PF Textbook: Tony Gaddis: Chapter 11
2	3	Structures with pointer as member Shallow Copy vs. Deep Copy	Your PF Textbook: Tony Gaddis: Chapter 11
	4	Arrays of Structures Struct within a struct	Your PF Textbook: Tony Gaddis: Chapter 11
3	5	Introduction to Object Oriented Concepts and Terminology: Real world examples	R2-(Ch-1 – Remedy-3) R2-(Ch-8)
	6	Define the keyword 'class' Access modifiers: private and public only; First Member function; Data Abstraction through Encapsulation; Setter/Mutator and Getter/Accessor methods; How the function knows which object invoked him? Answer will be given after a couple of lectures.	A-(Ch-3 (3.1~3.6))
4	7	Revision of Function Overloading & Default Arguments; Two Special Method: Constructor & Destructor; Constructor with No argument (Default Constructor); Constructor with Default arguments Calling sequence of Constructor & Destructor for multiple objects;	A-(Ch-3 (3.7~3.11)) A-(Chapter-9) A-(Chapter-10.5, 10.6)

		Pointer as data member Importance of destructors Pointer /Reference to objects, Passing objects to functions by reference Pointer this	
	8	Passing objects to functions by value, Default Member-wise copy (Assignment & Initialization), Problem of Member-wise copy & its solution: Copy Constructor, Calling sequence of Constructor & Destructor: when objects are passed by value/reference	
5	9	Preventing changes in data members from a method: const method; Constant data members Static functions Static data members constant and static objects Calling sequence of Constructor & Destructor for constant and static objects	A-(Chapter 10.2, 10.7)
	10	Object as data member (Composition), Aggregation; Cascading calls with and without this pointer.	A-(Chapter 10.3) R2-(Chapter 12)
6	11	Nameless objects Array of objects;	A-(Chapter 10.3) R2-(Chapter 12)
	12	Operator Overloading; Binary Operator receiving Instance of class as 1st operand	A-(Chapter 11.7)
7	13	Overloading Unary Operator: as member, as non-member Unusual Operators: ++, --,	A-(Chapter 11.11)
	14	[ ] (set & get, both versions), type-cast, Parentheses	A-(Chapter 11.8, 11.9)
8	15	Friend functions (efficient but shake the concept of encapsulation); Declaring a global function as friend of a class; Declaring member function of a class as friend of another class Friend class (an easy but more un-secure way);	A-(Chapter 10.4)
	16	Operator NOT receiving Instance of class as 1st operand Overloading Binary Operators (Stream Insertion/Extraction) for I/O stream	A-(Chapter 11.5)
Mid Term Examination			
9	17	In theory what is inheritance is-A Relationship: Public Inheritance Protected data member protected, private inheritance	A-(Chapter 12.1~12.4) R2-(Chapter 14)
	18	Multilevel Inheritance: Direct and Indirect Base Class Calling of Constructor and Destructor for Derived Class Objects; Explicit call to the constructor of Base class from Derived class;	A-(Chapter 12.5, 12.6) Case Study-A

10	19	Review of Simple Inheritance, Multi Level Inheritance, Multiple Inheritance A review of OO relationships terminologies: Aggregation, composition, generalization (is-A), knows-a.	A-(Chapter 12) Case Study-B
	20	Polymorphism: Overriding base-class members in derived class; Virtual functions and Dynamic binding; Concept of v-Pointer and v-Table	A-(Chapter 13) R1-Reading Material Case Study-B
11	21	Pure virtual functions and abstract class; Defining a pure virtual destructor	R1-Reading Material
	22	Detail discussion on v-table structure, virtual constructor, object cloning	R1-Reading Material
12	23	Some tricky things in pointer type casting and applying them on different topics studied so far.	R1-Reading Material
	24	Diamond inheritance Virtual inheritance	R1-Reading Material
13	25	C++ Streams, Members and Manipulators of Streams; File Handling using Streams	R1-Reading Material A-(Chapter Chapter-15)
	26	Access Techniques: Sequential, Direct, and Random Access Files; Input/Output of Object from/to File (binary/ text mode);	R1-Reading Material A-(Chapter Chapter-17)
14	27	Function Template; Overloading of Function Template; Specialized of Template Function	A-(Chapter 14.2,14.3)
	28	Class Template; Specialized method of Template Class; Complete Specialized Template Class	A-(Chapter 14.4, 14.5)
15	29	Friendship and Inheritance with Templates	A-(Chapter 14.6, 14.7, 14.8)
	30	What is Exception? Error vs. Exception; Evolution of Exception Handling: exit, abort, assert, new-keywords; try, catch, throw	A-(Chapter 16)
16	31	Unhandled Exception; Propagation of Exception and its advantage	A-(Chapter 16)
	32	Overall Review of Course for Final Exam	
Final Term Examination			

Web Sites:

1. Reference guide for C/C++ related stuff:
    - http://www.cplusplus.com/
    - http://www.parashift.com/c++-faq-lite/
    - http://c-faq.com/versions.html
    - The homepage of Bjarne Stroustrup, the inventor of C++: http://www.research.att.com/~bs

Code of Conduct

- In Quizzes/Tests, you are allowed to use any helping material available at that time unless specified otherwise. Neighbors and machines are exception.

- Things which surely lead to grade 'F'
    - Your neighbors are your enemies, so any sort of communication on assigned tasks will lead you to Grade 'F' in the current and in the previously submitted tasks.
    - Violation of coding convention.
    - Late Submissions.
    - Discussion or sniffing on neighbor's work in the laboratory/assigned tasks.

- Once the marks are published on Google docs for any graded task (sessional), You can question about any discrepancy about marks within *five* working days otherwise marks graded will be considered final.

- Mobile Phones must be switched off during the class and laboratory.

- How to Approach Me:
    - Observe the meeting hours!
      OR
    - Send an e-mail to Course E-mail
        - How to Send Email
            - Email Header/Subject
                - OOPF14
            - Email Body
                - Email Text must contain your roll-no and complete name