Republic of Iraq
The Ministry of Higher Education and Scintific Resrearch
Supervision and Scientific Evaluation Body

College : Shatt Al Arab University College
Department : Computer Technology Engineering
Stage: 2
Lecturer Name : Dr. Adel Nadhem Naeem
Academic Status :

Course Weekly Outline

Course Lecturer	**Dr. Adel Nadhem Naeem**				
e-mail	**Adel.naeem@sa-uc.edu.iq**				
Title	**Programming II**				
Course Coordinator					
Course Objective	**Introducing to the student the object-oriented programming and the use of the advanced C++ language as an example of object-oriented programming in resolve matters related to competence**				
Course Description	**•Introduce the student to variables, types of data, inputs and outputs** **•Recognizes and understands decision-making and logical, mathematical and conditional conditions** **•Recognizes and understands how to create conditions and loops and how to continue and stop** **•Recognize and understand the functions and all the identifications and definitions related to them etc.** **•Recognizes and understands matrices with one dimension and two dimensions**				
Textbook	**"Object-Oriented Programming in C++"**, 4th Edition, Robert Lafore, Sams Publishing, 2002.				
References	**"CPA: Programming Essentials in C++"**, C++ INSTITUTE, 2016. **"C++ Tutorial"**, tutorialspoint. https://www.tutorialspoint.com/cplusplus/index.htm				
Course Assessment	**Term Exam**	**Lab**	**Quizzes and Attendance**	**Project**	**Final Exam**
	30	**10**	**10**	**-**	**50**
General Notes					

Republic of Iraq
The Ministry of Higher Education and Scintific Resrearch
Supervision and Scientific Evaluation Body

College : Shatt Al Arab University College
Department : Computer Technology Engineering
Stage: 2
Lecturer Name : Dr. Adel Nadhem Naeem
Academic Status :

Week	Class Subjects	Lab Subjects	Goals
1 - 2	C++ Review (Program structure, namespace, identifiers, variables, constants, enum, operators, typecastings, control structures and functions).	C++ Review (Program structure, namespace, identifiers, variables, constants, enum, operators, typecastings, control structures and functions).	The learner should be able to recall the basic concepts and tools of structural programming using the C++ language
3	Introduction to Object-Oriented Programming in C++.	Introduction to Object-Oriented Programming in C++.	The student should be able to understand the basic concepts of object-oriented programming
4 - 8	Objects and Classes (Basics of objects an classes in C++, private and public members, static data and function members, constructors and their types, destructors and operator overloading).	Objects and Classes (Basics of objects an classes in C++, private and public members, static data and function members, constructors and their types, destructors and operator overloading).	The student should be able to analyze, design and implement software solutions to applied problems according to the concepts of object-oriented programming
9 - 14	Inheritance (Concepts of Inheritance, types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members, overriding, virtual base class).	Inheritance (Concepts of Inheritance, types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members, overriding, virtual base class).	The student should be able to apply the concepts of heredity in the programs he builds to achieve the greatest possible reduction in the code
15 - 19	Polymorphism (Pointers in C++, Pointes and Objects, this pointer, virtual and pure virtual functions, Implementing polymorphism).	Polymorphism (Pointers in C++, Pointes and Objects, this pointer, virtual and pure virtual functions, Implementing polymorphism).	The learner should be able to understand, design and apply programmatic problems that include the concept of polymorphism
20 - 24	I/O and File	I/O and File management	The student should be able to

	management (Concepts of streams, cin and cout objects, C++ stream classes, Unformatted and formatted I/O, manipulators, File stream, C++ File stream classes, File management functions, File modes, Binary and random files).	(Concepts of streams, cin and cout objects, C++ stream classes, Unformatted and formatted I/O, manipulators, File stream, C++ File stream classes, File management functions, File modes, Binary and random files).	deal with files in their various forms to store and retrieve data
25 - 30	Templates, Exceptions and STL (What is template? function templates and class templates, Introduction to exception, try-catch-throw, multiple catch, catch all, rethrowing user defined exceptions, Overview and use of Standard Template Library).	Templates, Exceptions and STL (What is template? function templates and class templates, Introduction to exception, try-catch-throw, multiple catch, catch all, rethrowing user defined exceptions, Overview and use of Standard Template Library).	The student should be able to develop general programs that do not depend on a specific type of data as well as deal with algorithms and general data structures commonly used, and be able to design programs that have the ability to deal with error situations that occur during the implementation of the program

Lecturer signature **Head of Department Signature**