College of Computer and Information Sciences
كلية علوم الحاسب والمعلومات

Kingdom of Saudi Arabia
Ministry of Higher Education
College of Computer & Information Sciences
Majmaah University

جامعة المجمعة
Majmaah University

Course Profile

Course Name:-	**Programming II (Object oriented Programming)**
Course Code:-	**CS 120**
Academic Year:-	**1434-1435(H)**
Semester:-	**I**

Course Overview

This course provides in-depth coverage of object-oriented programming principles and techniques using C++. Students develop fundamental programming skills using a language that supports an object-oriented approach, incorporating security awareness, human-computer interactions and social responsibility. Object-oriented software development has become a standard methodology throughout the software engineering discipline. Therefore, a solid grasp of object-oriented programming is essential for any Computer Science /information technology specialist. While there are a variety of object-oriented programming languages available, C++ is one of the most widely used and is therefore the focus of this course. This course provides in-depth coverage of object-oriented programming principles and techniques using C++. This course introduces the object-oriented programming concepts, principles, and techniques, including classes, objects, inheritance, and polymorphism. All these concepts are illustrated via a contemporary object-oriented programming language. Topics include classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes, in the context of the C++ language.

Course Details	
Level:-	4
Credit:-	**4**
Pre-Requisites:-	CS 110
Co- Requisites:-	**NA**

Learning Outcomes of Course

On successful completion of this course, you will be able to:

1. Comprehend what programming is and what a programming language does; Know about the evolution of C++;
2. Identify and design suitable classes and class hierarchies and code class implementations in C++;
3. Design and develop C++ programs using classes, constructors and destructors, static data members and static member functions, reference variables,
4. Apply the principles of inheritance, polymorphism , function overloading and operator overloading in C++ programs
5. Apply the principles of dynamic memory allocation & virtual functions in C++ programs.
6. Design and develop C++ programs using file processing ,templates and C++ standard library (STL)

Course Assessment

Name of Assessment Task	Weight of Assessment	Week Due
Midterm Exam-1	15%	**6**
Midterm Exam-2	15%	**12**
Quizzes	5%	**4,8,13**
Assignments/Report/Seminar	5%	**3,5,10,11,14**
Lab Exam	20%	**15**
Final Exam	40%	**16**

Assessment Task and Learning Outcomes Alignment

Assessment Task Name	Course Learning Outcomes					
	1	**2**	**3**	**4**	**5**	**6**
1. Midterm Exam-1	√	√	√			
2. Midterm Exam-2				√	√	
3. Quizzes	√	√	√	√	√	
4. Assignments/Report/Seminar	√	√	√	√	√	√
5. Lab Exam	√	√	√	√	√	√
6. Final Exam	√	√	√	√	√	√

Teaching Contact Details

Course Coordinator:	**Shailendra Mishra, Ph.D**
Lab/Tutorial Instructor:	**Abdul Rahim Khan ,M.Tech**
Email:	s.mishra@mu.edu.sa m.khan@mu.edu.sa
Office Hours:	Thursday 10 AM-12 NOON
Office Number:	0164045382
Office:	Level 1, CCIS Building Room No-024-1-20-3

Details of Required Text Book

Book Name	Authors Name	Publisher	Year	Edition
1. C++: How To Program	Paul Deitel and Harvey Deitel	Prentice Hall	**2012**	**8th**

Details of Required Reference Books

Book Name	Authors Name	Publisher	Year	Edition
1. The C++ Programming Language: Special Edition,	Bjarne Stroustrup	Addison-Wesley	2013	**4th**
2. C++ Programming: From Problem Analysis to Program Design	D. S. Malik	Course Technology	2012	**6 th**
3. C++ Programming for the Absolute Beginner	Dirk Henkemans and Mark Lee	Thomson Course, Technology	2007	

IT Resources

The following IT Resources will require to access-

- C++ quick reference
- cplusplus.com - The C++ Resources Network
- Free Programming Resources dot com
- Programming Tutorials dot com
- C++ Made Easy
- POSIX Reference from IEEE Open Group Base Specification

Course Schedule

Module/Topic	Chapter	Event and submission	Week
Classes and Objects • Structure Definition • Classes vs. Objects • Interface and Implementation	Introduction to class object and String ,Chapter III, Paul Deitel and Harvey Deitel	Brain storming and review of previous knowledge.	Week-1
• Constructors and Destructors • Set and get Functions • Constant Objects and constant Member	Introduction to class object and String ,Chapter III, Paul Deitel and Harvey Deitel		Week-2
Functions • Constant parameters, Constant return types • Friend Functions and friend Classes • Static Class Members • Composition	Function & Introduction to recursion Chapter V, Paul Deitel and Harvey Deitel	Assignment I	Week-3
• Dynamic Memory Management (creating object at run-time) • Arrays of objects	Arrays & Vector,Chapter VII,Pointers Chapter VIII, Paul Deitel and Harvey Deitel	Quiz 1	Week-4
Inheritance • Base Classes and Derived Classes • protected Member • public, protected and private Inheritance	Object oriented programming-Inheritance Chapter XII, Paul Deitel and Harvey Deitel	Assignment II	Week-5
		Written Assessment Due	Week-6
• Inheritance Hierarchy • Software reusability using Inheritance • Multiple inheritance • Virtual Functions	Object oriented programming-Inheritance Chapter XII, Paul Deitel and		Week-7

• Abstract classes and pure virtual function	Harvey Deitel		
Polymorphism • Relationships Among Objects in an Inheritance Hierarchy • Invoking Base-class Functions from Derived Class Objects • Aiming Derived-Class Pointers at Base Class Objects	Object oriented programming-Polymorphism Chapter XIII, Paul Deitel and Harvey Deitel	Quiz II	Week-8
• Derived-Class Member-Function Calls via Base-Class Pointers **Operator Overloading** • Fundamentals of operator overloading	Object oriented programming-Polymorphism Chapter XIII, Paul Deitel and Harvey Deitel Operator Overloading-String Chapter XI, Paul Deitel and Harvey Deitel		Week-9
• Restrictions of operator overloading • Global and member operators • Overloading Stream-Insertion and Stream-Extraction Operators • Overloading Unary Operators (++,--,! Etc..) • Overloading Binary Operators(+,-,* etc..)	Operator Overloading-String Chapter XI, Paul Deitel and Harvey Deitel	Assignment III	Week-10
File processing • Files and Streams classes in C++ • Creating a Sequential File Reading Data from a Sequential File	File processing Chapter XVII, Paul Deitel and Harvey Deitel	Assignment VI	Week-11
		Written	Week-12

		Assessment Due Sunday	
• Updating Sequential Files • Random-Access Files • Creating a Random-Access File • Writing and updating Random-Access File	File processing Chapter XVII, Paul Deitel and Harvey Deitel	Quiz III	Week-13
Templates • Function Templates • Class Templates • Containers and templates	Templates Chapter XIV, Paul Deitel and Harvey Deitel	Assignment V	Week-14
			Review Exam Week
			Exam Week

The **American Psychological Association (APA)** referencing style must be use for all submissions of this course.

Course Assessment Task

WRITTEN ASSESMENT (Mid Term I Exam)

Assessment Title	Written Assessment
Task Description	This assignment is aligned to learning outcomes 1, 2,3 In that regard, the assignment contains questions that assess: 1. Comprehend what programming is and what a programming language does; Know about the evolution of C++; 2. Identify and design suitable classes and class hierarchies and code class implementations in C++; 3. Design and develop C++ programs using classes, constructors and destructors, static data members and static member functions, reference variables, function overloading and operator overloading & inheritance. The complete details of the assessment task are provided in Module.
Assessment Due Date	**Week 6**
Return Date to Students	Week 8 Thursday
Weighting	15%
Assessment Criteria	The assessment criteria for this task are under continuous revision.
Referencing Style	American Psychological Association (APA)
Submission	
Learning Outcomes Assessed	1. Comprehend what programming is and what a programming language does; Know about the evolution of C++; 2. Identify and design suitable classes and class hierarchies and code class implementations in C++; 3. Design and develop C++ programs using classes, constructors and destructors

WRITTEN ASSESMENT (Mid Term II Exam)

Assessment Title	Written Assessment
Task Description	This assignment is aligned to learning outcomes 3, 4,In that regard, the assignment contains questions that assess: • Apply the principles of inheritance, polymorphism , function overloading and operator overloading in C++ programs • Apply the principles of dynamic memory allocation & virtual functions in C++ programs. The complete details of the assessment task are provided in Module.
Assessment Due Date	**Week 12**
Return Date to Students	Week 13
Weighting	15%
Assessment Criteria	The assessment criteria for this task are under continuous revision.
Referencing Style	American Psychological Association (APA)
Submission	
Learning Outcomes Assessed	• Apply the principles of inheritance, polymorphism , function overloading and operator overloading in C++ programs • Apply the principles of dynamic memory allocation & virtual functions in C++ programs.

FINAL EXAMINATION

Outline	**Complete an examination**
Date	**During University examination period**
Weighting	**40%**
Length	180 Minutes
Details	Dictionary - non-electronic, concise, direct translation only (dictionary must not contain any notes or comments) No Calculator Permitted Closed Books
Learning Outcomes Assessed	1. Comprehend what programming is and what a programming language does; Know about the evolution of C++; 2. Identify and design suitable classes and class hierarchies and code class implementations in C++; 3. Design and develop C++ programs using classes, constructors and destructors, static data members and static member functions, reference variables, 4. Apply the principles of inheritance, polymorphism , function overloading and operator overloading in C++ programs 5. Apply the principles of dynamic memory allocation & virtual functions in C++ programs. 6. Design and develop C++ programs using file processing ,templates and C++ standard library (STL)