

CS 120

Programming II (Object oriented Programming)

Term 2 - 2014

Course Profile

All details in this course profile for CS120 have been officially approved by CCIS Majmaah University and represent a learning partnership between the University and you (our student). The information will not be change unless absolutely necessary and any change will be clearly indicated by an approved correction included in the profile.

General Information

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.

DETAILS

Level	4
Credit Points	3

PRE-REQUISITES OR CO-REQUISITES

Pre-requisite: CS 110

ATTENDANCE Requirements

All CSIS students are expected to attend scheduled classes, these classes are identified as a mandatory (pass/fail) component and attendance is compulsory. The attendance and academic progress requirements in each study period (satisfactory attendance for all students is defined as maintaining at least a 75 % attendance record).

ASSESSMENT OVERVIEW

Assessment Task	Weighting
1. Midterm Exam-1	15%
2. Midterm Exam-2	15%
3. Quizzes	5%
4. Assignments/Report/Seminar	5%
5. Lab	20%
6. Final Exam	40%

This is a graded course: your overall grade will be calculated from the marks or grades for each assessment task, based on the relative weightings shown in the table above. You must obtain an overall mark for the course of at least 60%, or an overall grade of 'pass' in order to pass the course. If any 'pass/fail' tasks are shown in the table above they must also be completed successfully ('pass' grade). You must also meet any minimum mark requirements specified for a particular assessment task, as detailed in the 'assessment task' section (note that in some instances, the minimum mark for a task may be greater than 60%). Consult the University's Grades and Results Procedures for more details of interim results and final grades.

Majmaah University Policies

All University policies are available on the mu.deu.sa.

You may wish to view these policies:

- Assessment of Coursework Procedures
- Grads and Results Procedure
- Review or Grade Policy
- Plagiarism Procedure
- Student Misconduct and Plagiarism Policy
- Monitoring Academic Progress Policy
- Monitoring Academic Progress Policy
- Monitoring Academic Progress Procedures
- Refund Excess Payments (Credit Balances) Policy
- Student complaints Policy
- Use of Internet, mail and Computing Facilities Policy

This list is not an exhaustive list of all University policies. The full lists of University policies are available on the University Web site (www.mu.edu.sa)

Course Learning outcomes

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)

Alignment of Learning outcomes, Assessment and Graduate attributes

ALIGNMENT OF ASSESSMENT TASKS TO LEARNING OUTCOMES

Assessment Task	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

Textbook and Resources

PRESCRIBED TEXTBOOKS

C++: How To Program, , 8th edition, , 2012			
Author/s	: Paul Deitel and Harvey Deitel	Year	: 2012
Edition	: 8th	Publisher	: Prentice Hall

IT RESOURCES

You will need access to the following IT resources:

- [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

Referencing style

All submissions for this course must use the **American Psychological Association (APA)** referencing. For further information, see the Assessment Tasks below.

Teaching Contacts

Course Coordinator	Prof. Shailendra Mishra College of Computer & Information Sciences Majmaah University, Majmaah, KSA
---------------------------	--

Schedule

Week	Module/Topic	Chapter	Event and submission
Week-1	Classes and Objects <ul style="list-style-type: none"> • 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-2	<ul style="list-style-type: none"> • 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-3	Functions <ul style="list-style-type: none"> • 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-4	<ul style="list-style-type: none"> • 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-5	Inheritance <ul style="list-style-type: none"> • 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
Vacation week			
Week-6			Written Assessment Due Sunday (16 March 2014)

			10:00 PM
Week-7	<ul style="list-style-type: none"> • Inheritance Hierarchy • Software reusability using Inheritance • Multiple inheritance • Virtual Functions • Abstract classes and pure virtual function 	Object oriented programming- Inheritance Chapter XII, Paul Deitel and Harvey Deitel	
Week-8	Polymorphism <ul style="list-style-type: none"> • 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-9	<ul style="list-style-type: none"> • Derived-Class Member-Function Calls via Base-Class Pointers Operator Overloading <ul style="list-style-type: none"> • 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-10	<ul style="list-style-type: none"> • 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-11			Written Assessment Due Sunday (20 April 2014) 10:00 AM

Week-12	File processing <ul style="list-style-type: none"> 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-13	<ul style="list-style-type: none"> 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-14	Templates <ul style="list-style-type: none"> Function Templates Class Templates Containers and templates 	Templates Chapter XIV, Paul Deitel and Harvey Deitel	Assignment V
Review Exam Week			
Exam Week			

Teaching Contacts

Contact information

Course Coordinator:	Shailendra Mishra, Ph.D
Lab/Tutorial Instructor:	
Email:	s.mishra@mu.edu.sa
Office Hours:	8.00 a.m. to 02.30 p.m.
Office Number:	0164045382
Office:	Level 1, CCIS Building Room No-3-2-20-2 Majmaah University

For any individual queries, please email me and I will endeavour to reply as soon as practical.

Assessment Task

WRITTEN ASSESMENT(Mid Term I Exam)

Assessment Title	Written Assessment
Task Description	<p>This assignment is aligned to learning outcomes 1, 2,3 In that regard, the assignment contains questions that assess:</p> <ol style="list-style-type: none">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. <p>The complete details of the assessment task are provided in Module.</p>
Assessment Due Date	Week 6 Sunday (16 March 2014)10 :00 AM
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	<ol style="list-style-type: none">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	<p>This assignment is aligned to learning outcomes 3, 4, In that regard, the assignment contains questions that assess:</p> <ul style="list-style-type: none">• 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. <p>The complete details of the assessment task are provided in Module.</p>
Assessment Due Date	Week 11
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	<ul style="list-style-type: none">• 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	<ol style="list-style-type: none">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++ programs5. 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)