
OBJECT- ORIENTED PROGRAMMING

School of Advanced Technology

Course Number: NET2006	Contribution to Program: Core	Educator(s): Rex Woollard
Applicable Program(s): Bachelor of Information Technology – Network Technology	AAL: 03	Approved For: Fall 2008
Course Hours: 5 contact hours per week	Prerequisites: NET1000	Approved By: Shawn McBride, Acting Chair Computer Studies Department
	Corequisites: None	Approved for Academic Year: 2008-2009

COURSE DESCRIPTION

Study of Object-Oriented Programming principles, emphasizing the development of efficient and reusable systems. Topics include: encapsulation, polymorphism, overloading, memory management, exception handling, and design templates and libraries.

COURSE CURRICULUM

Course Learning Requirements/Embedded Knowledge and Skills

Course Learning Requirements	Knowledge and Skills
<p>To earn credit for this course, you must reliably demonstrate your ability to:</p>	
<p>1. Use object-oriented programming principles and techniques to devise a well-defined solution to a problem.</p>	<ul style="list-style-type: none"> • Explain object-oriented concepts and techniques, and implement C++ solutions for the following: <ul style="list-style-type: none"> • classes • nested classes • C++ strings (compared to C- style strings) • operator overloading (to support the use of the STL) • inheritance • virtual functions (and the associated use of inheritance and pointers) • exceptions and exception handling • templates and Standard Template Library (STL) • Implement advanced data types: pointer to object, array of objects, pointer to array of objects, pointer to an array of pointers to objects, strings, abstract data types such as STL containers. Explain the reasons for using different approaches to data management. • Create code based on UML OO diagrams, and create UML OO diagrams to describe the OO organization of a solution to a given problem. • Explain and implement sequential file handling techniques.
<p>2. Write, compile, execute, and debug C++ language programs using classes.</p>	<ul style="list-style-type: none"> • Use C++ syntax to create working programs for a variety of applications. • Write effective, compact, and correct C++ language programs. • Use a debugger to trace program execution as a technique to prove correctness and as a technique to identify the source of program errors.

Course Learning Requirements	Knowledge and Skills
<p>To earn credit for this course, you must reliably demonstrate your ability to:</p>	
<p>3. Develop appropriate documentation of the solution to a problem</p>	<ul style="list-style-type: none"> • Develop UML diagrams to document class structures and composition. • Develop use-cases and comprehensive test plans.

Learning Resources

Optional Textbook:

- Object-Oriented Programming in C++ 4th ed., by Robert Lafore, Sams, ISBN 0-672-32308-7

Additional Materials:

Weekly theory notes, multi-media lectures and examples, quizzes and assignments available through web-based repositories.

I. Teaching/Learning Methods

The course consists of 3 hours of lectures and 2 hours of labs per week. You will likely need to spend an additional 5 hours per week, on average, of your own time for assignments and study.

II. Learning Activities

Samples of learning activities include:

- classroom lectures
- laboratory work
- practical and reading assignments
- research of course-related material

III. Course Content

Lectures

- Theoretical course material will be presented, aided by use of overhead projections, videos, demonstrations and brief lecture notes.
- Students are expected to attend all of the lectures and review all of the materials posted in web-based repositories.
- Students will be expected to find and read applicable material in the textbook, and to be prepared to answer oral or written questions in lectures.
- Students are encouraged to ask questions during lectures and to consult with the professors on topics that they do not clearly understand. Students are also encouraged to ask/answer questions or initiate discussions using the Web based Discussion Board forums. Professors will inform students, at the beginning of the course, of suitable times for consultations.

Labs

- Students will apply the lecture material to a series of assignments which are closely integrated with the current lecture materials.
- The students' ability to successfully complete the assigned exercises will directly correlate with their level of success on tests and the final exam.
- Labs build on earlier work, and are increasingly complex as the course progresses.
- Students should seek advice and help from the professors in the laboratory. Students who have been attending labs may still need extra help from the professor outside of the scheduled lab period. In such cases, students can use the online Discussion Board forums or make an individual appointment to see the professor.
- Lab assignments, online quizzes and exercises can be completed on college computers or at home if students have appropriate equipment and software.

Planned Topics and Pacing

Weeks	Topics	Text Reference
Weeks 1 & 2	<ul style="list-style-type: none"> • Problem Decomposition • Test Plan Development • Data types, keyboard input and screen output, library functions, and type conversion • Control structures - sequential, selection, and repetition • Aggregate data structs • Bit-level data manipulation • Decompose and modularize with functions. • Communicate with functions using <i>pass-by-value</i>, <i>pass-by-reference</i>). • Use of debugger to explore <i>pass-by-value</i>, <i>pass-by-reference</i> techniques. • File processing: using <i>ifstream</i>, <i>ofstream</i> objects, methods, <i>open()</i>, <i>fail()</i>, <i>good()</i>, <i>close()</i>, operators <i>>></i>, <i><<</i> and <i>getline()</i> • Passing <i>stream</i> objects to functions (<i>pass-by-reference</i>). 	Chapter 2 Chapter 3 Chapter 4 Chapter 5 Chapter 12
Weeks 3 to 5	<ul style="list-style-type: none"> • The Object-Oriented Approach: encapsulation, data hiding, overloading, polymorphism, inheritance. • Classes: data/actions, declaration, constructors, functions (methods), declaring / defining objects, executing constructors and methods, overloading functions. • Role of member access operator. The hidden <i>this</i> pointer in function calls. • Static class variables and static class functions (methods). • Parsing a problem description for nouns, verbs and instances of <i>is-a</i> and <i>has-a</i> relationships • Draw static class UML diagrams (using Visio) • Draw memory maps of objects at run-time with sample data (using Visio) 	Chapter 1 & Chapter 6

	<ul style="list-style-type: none"> • <i>string</i> Class (C++-style strings): simple, intuitive use only. 	
Weeks 6 & 7	<ul style="list-style-type: none"> • Pointers: <ul style="list-style-type: none"> ○ & (address of), ○ * (dereference), ○ old c-style passing info in parameter list ○ relationship to arrays • Dynamic Memory Allocation: <ul style="list-style-type: none"> ○ use of <i>new</i> and <i>delete</i> ○ implementation in a class ○ destructors ○ copy constructors • Use the debugger to inspect the organization of memory when using pointers to dynamically allocated objects. 	Chapter 7 & Chapter 10
Week 8	<ul style="list-style-type: none"> • Inheritance: base classes and derived classes 	Chapter 9
Week 9	<ul style="list-style-type: none"> • Operator Overloading (overloading == and < to support the use of the STL – no << >> overloading) 	Chapter 8
Week 10	<ul style="list-style-type: none"> • Virtual Functions and the role of inheritance and pointers. 	Chapter 11
Week 11	<ul style="list-style-type: none"> • Class templates (understand the syntax, only use pre-existing template classes), Standard Template Library (STL): <i>containers</i>, <i>iterators</i> and <i>algorithms</i>. 	Chapter 14
Week 12	<ul style="list-style-type: none"> • Exceptions, exception handling, syntax with arguments 	Chapter 14

Evaluation/Earning Credit

The following will provide evidence of your learning achievement:

Assessment of student learning will be done by means of class tests, final exam and laboratory assignments. Class tests may consist of scheduled in class tests and/or scheduled compulsory weekly quizzes.

The Computer Studies Department requires that all course assignments (homework exercises, laboratory work, projects, etc) be submitted by students using a standard which could be specific to one or more courses. Professors will ensure, at the beginning of the term, that students are advised of the exact details of these course specific submission requirements. Professors will also post them online alongside the course outline. Student submissions that do not meet the course published submission standards may not be marked, and may incur a penalty of up to 100% of the submission mark.

The factors in the final grade are:

Classroom test(s), quizzes **30%**

Final examination **40%**

The final exam is required and designed to be a comprehensive, summative evaluation instrument.

Assignments **30%**

- Late submissions will receive a grade of zero. Partially completed lab work may be eligible for significant partial marks and should be presented for evaluation in spite of its unfinished state.
- In the case of a documented emergency the professor, in consultation with the Chair, will determine how the marks will be made up and/or final grade adjusted.

In order to obtain a credit in this course, students must achieve a minimum of 50% from all the evaluation factors combined. Note however that at least 50% must be achieved in the combined grade for tests and the final examination (i.e. 35 out of 70) in order for laboratory assignments to be added into the final grade.

All students are required to write the final exam. If, as a result of being off-track in your program, you note that there is a scheduling conflict in your final exam schedule, it is your responsibility to alert your course professor no later than one week before the start of the final exam period, to allow for any special arrangements. For any other situations resulting in a student not writing their final exam, the normal Carleton University rules for missed final exams will apply. See <http://www.carleton.ca/ses/deferred/index.html> for details.

Related Information

Retention of course material. It is your responsibility to retain copies of all assignments, labs and mid-term tests (returned from the professor), and any other evaluations and pertinent records (except for final exams, which are not returned) in case you become involved in an appeal hearing at a later date.

It is also your responsibility to retain course outlines for possible future use to support applications for transfer of credit to other educational institutions.

College email account. Algonquin College provides all full-time students with an email account. This is the address that will be used when the College, your professors, or your fellow students communicate important information about your program or course events. It is your responsibility to ensure that you know how to send and receive email using your Algonquin College account, and to check it regularly.

Harassment/Discrimination/Violence will not be tolerated. Any form of harassment (sexual, racial, gender or disability-related), discrimination (direct or indirect), or violence, whether involving a professor and a student or amongst students, will not be tolerated on the college premises.

Harassment means one or a series of vexatious comment(s) (whether done verbally or through electronic means), or conduct related to one or more of the prohibited grounds that is known, or ought reasonably to be known, to be unwelcome/unwanted, offensive, intimidating, derogatory or hostile.

This may include, but is not limited to: gestures, remarks, jokes, taunting, innuendo, display of offensive materials, offensive graffiti, threats, verbal or physical assault, stalking, slurs, shunning or exclusion related to the prohibited grounds.

Bachelor of Information Technology students are bound by the “Academic Regulations of the University – Student Conduct”, “15. Offences of Conduct: Discrimination and Harassment” detailed within Carleton University’s Undergraduate Calendar, and online at:

<http://www.carleton.ca/cuuc/regulations/acadregsuniv15.html>

The School of Advanced Technology’s Standard Operating Procedure on Plagiarism and Academic Honesty defines plagiarism as an attempt to use or pass off as one’s own idea or product, work of another without giving credit. Plagiarism has occurred in instances where a student either directly copies another person’s work without acknowledgement; or, closely paraphrases the equivalent of a short paragraph or more without acknowledgement; or, borrows, without acknowledgement, any ideas in a clear and recognizable form in such a way as to present them as one’s own thought, where such ideas, if they were the student’s own would contribute to the merit of his or her own work.

Plagiarism is one of the most serious academic offences a student can commit.

Bachelor of Information Technology students are bound by the “Academic Regulations of the University – Student Conduct”, “14. Instructional Offences” detailed within Carleton University’s Undergraduate Calendar, and online at: <http://www.carleton.ca/cuuc/regulations/acadregsuniv14.html>

Violation of the Copyright Act.

- **General** – The Copyright Act makes it an offence to reproduce or distribute, in whatever format, any part of a publication without the prior written permission of the publisher. For complete details, see the Government of Canada website at <http://www.cb-cda.gc.ca/info/act-e.html>. Make sure you give it due consideration, before deciding not to purchase a textbook or material required for your course.
- **Software Piracy** - The Copyright Act has been updated to include software products. Be sure to carefully read the licensing agreement of any product you purchase or download, and understand the term and conditions covering its use, installation and distribution (where applicable). Any infringement of licensing agreement makes you liable under the law.

The Use of Electronic Devices, during classes, other than those sanctioned by the course professor is strictly prohibited. In particular, cell phones are not to be used to communicate during a class. The use of any electronic devices during exams and mid-term tests, other than those sanctioned by the faculty in charge of the examination is strictly prohibited.

In accordance with College Directive E39, any unauthorized use of a prohibited device will be considered plagiarism, and be dealt with as such. In these cases, Bachelor of Information Technology students would be bound by the "Academic Regulations of the University - Student Conduct", "14. Instructional Offences" detailed within Carleton University's Undergraduate Calendar, and online at: <http://www.carleton.ca/cuuc/regulations/acadregsuniv14.html>

Disruptive Behaviour is any conduct, or threatened conduct, that is disruptive to the learning process or that interferes with the well being of other members of the College community. It will not be tolerated.

Members of the College community, both students and staff, have the right to learn and work in a secure and productive environment. The College will make every effort to protect that right.

Incidents of disruptive behaviour must be reported in writing to the departmental Chair as quickly as possible. The Chair will hold a hearing to review available information and determine any sanctions that will be imposed. Disciplinary hearings can result in penalties ranging from a written warning to expulsion.

For further details, consult Algonquin College Directive – E27, Instaguide.

Students with Disabilities requiring academic accommodations in this course are encouraged to contact a coordinator at the Paul Menton Centre (PMC) for Students with Disabilities to complete the necessary *letters of accommodation*. After registering with the PMC, make an appointment to meet and discuss your needs with the professor at least two weeks prior to the first in-class test or instructional television midterm exam. This is necessary to ensure sufficient time for making any needed arrangements. Please note the deadline for submitting completed forms to the PMC as published in Carleton University’s “Academic Year” calendar.

Challenge for Credit

Challenge for credit is a Carleton University policy that enables students to obtain undergraduate academic credit for any learning and experience gained through work and related professional development. It is not intended to overlap in scope with transfer of credits or admission with advanced standing.

For full details, see Carleton University's Undergraduate Calendar, "Academic Regulations of the University", Section 1.9, also available online at: <http://www.carleton.ca/cu0809uc/regulations/acadregsuniv1.html#1.9>

For this course, candidates will provide evidence of their learning achievement through the successful completion of:

- A challenge exam with a breadth of coverage and level of difficulty equivalent to the final examination in the course; plus,
- A hands-on or practical component to demonstrate the achievement of the requisite applied knowledge and skills.

Eligibility for Deferred Examination

Only students who have achieved satisfactory performance during the term will be eligible for a deferred examination. In accordance with the factors determining the final grade in section VI above, satisfactory performance leading to the final exam is defined as the student having achieved 50% in all aspects of the course marking scheme (save the final exam), be they grouped (e.g. practical component, lab component, theory component, etc) or individually listed.

Students who have failed the course on the basis of inadequate term course work will receive a grade of FND – failure with no deferred final examination allowed.