|  |
| --- |
| **Course Title: SEP200 Object-Oriented Programming Year and Semester:** Year 1, Semester 2 |
| **Course Description**  Students expand their knowledge of programming fundamentals with the C++ language by examining the fundamentals of object-oriented programming – encapsulation, inheritance, and polymorphism. An introduction to the standard template library is given, as well as an introduction to UML. The principles of object- oriented programming presented here are applicable to a wide variety of languages  and are fundamental to modular software development within a large code base. |
| **Prerequisite(s) or co-requisite(s)**  SEP100 |
| **Method of Instruction**   * Lecture * Laboratory * Online |
| Content Outline by Topic   * Complexity, Languages, Namespaces * Foundations * Encapsulation * Inheritance * Polymorphism * Templates * Class Relationships * Unified Modelling Language * Memory Model * Standard Template Library * Language Standards |
| **Actual Contact Hours/Week**  4 hours a week for 14 weeks |
| **Methods and Frequency of Evaluation of Student Performance**  Workshops (8 minimum) 30%  Final Project 20%  Quizzes (minimum of 8) 10%  Test 20%  Final Exam 20% |
| **Resources to be Purchased/Provided by Students**  None |
| **Textbook Requirement:**  Introduction to Object-Oriented Programming Using C++11 (July 2016 Edition) by Chris M. Szalwinski  Seneca College  Available in the bookstore and online  The C++ Programming Language (4th Edition) by Bjarne Stroustrup |

|  |
| --- |
| Addison-Wesley Professional ISBN-10: 0321563840  ISBN-13: 978-0321563842  http://libaccess.senecacollege.ca/login?url=https://[www.safaribooksonline.com/library/](http://www.safaribooksonline.com/library/) view/the-c-programming/9780133522884/?ar&orpq&email=^u  C++ Primer (5th Edition) by Stanley B. Lippman ISBN-10: 9780321714114  ISBN-13: 978-0321714114  http://libaccess.senecacollege.ca/login?url=https://[www.safaribooksonline.com/library/](http://www.safaribooksonline.com/library/) view/c-primer-fifth/9780133053043/?ar&orpq&email=^u |
| **Learning Outcomes**  *Upon successful completion of this course the student will be able to:*   1. Design classes to model the components of a programming solution. 2. Design member functions using logic constructs to solve tasks of linear complexity. 3. Use stream objects to interact with users and access persistent data. 4. Trace the execution of object-oriented code to validate its correctness. 5. Develop an algorithm using object-oriented concepts to solve a simple programming problem. 6. Model generalization and specialization using single and multiple inheritance to minimize the duplication of code in complex hierarchies. 7. Model polymorphic behavior using interfaces, virtual functions and templates (generics) to amplify the reusability of code. 8. Implement design components using algorithms of the standard template library to utilize existing technologies. 9. Design program components using raw pointers, pointer arithmetic and smart pointers to access data in program memory. |
| **Faculty Qualifications to Teach this Course**  Masters or PhD in computer science, or software/electrical engineering or equivalent |
| **Faculty Qualified to Teach/Supervise this Course**  Robert Robson, PhD |
| **Percentage of Course Content Offered Online**  25% Interactive content online |
| **Classroom Requirements**  ☒Regular classroom   * Electronic classroom   ☒Computer Lab   * Activity-based Learning centre |
| **Equipment Requirements**  College computers are configured to support this course. If you use your own computer, you will need:   * Current version of an operating system (Mac OS X, Windows, Linux), with Internet   access |

|  |
| --- |
| * Software development tools, including text and/or programmer's editor(s), and compiler(s) |