1. **Course number and name**
   CMP 305 – Data Structures and Algorithms

2. **Credits and contact hours**
   3 credit hours, 4 contact hours

3. **Instructor's or course coordinator's name**
   Dr. Michel Pasquier

4. **Textbook, title, author, and year**
   N. Dale. *C++ Plus Data Structures,* 5th edition, Jones and Bartlett, 2013.

   Other Supplemental Materials
   M. Goodrich, D. Mount, R. Tamassia. *Data Structures and Algorithms in C++*, 2nd edition, Wiley and Sons, 2011.

5. **Specific course information**
   a. **Brief description of content of the course (catalog description)**
   Covers design and implementation of abstract data types, lists, stacks, queues and trees. Covers recursion and runtime stacks. Introduces the complexity of algorithms and data structures. Covers searching and sorting and basic graph algorithms. Laboratory work includes substantial programming assignments.

   b. **Prerequisites or co-requisites**
   Prerequisites: CMP 220 (Introduction to Computer Science II) or COE 211 (Programming II)
   Prerequisites/Concurrent: CMP 213 (Discrete Structures) or MTH 213 (Discrete Mathematics)

   c. **Indicate whether a required, elective, or selected elective course in the program**
   Required

6. **Specific goals for the course**
   a. **Specific outcomes of instruction**
   This course requires the student to demonstrate the following:
   1. Derive and use the time complexity measure to compare the performance of basic algorithms.
   2. Specify and implement Abstract Data Types (ADTs) using an object-oriented programming language.
   3. Develop software applications that make use of one or more ADTs, including stacks, queues, lists, binary trees, search trees, heaps and graphs, comparing and selecting different designs based on performance criteria.
   4. Use recursion to implement data structure algorithms.
   5. Use sorting and searching, including sequential, binary and hash tables to solve real-world problems.

b. Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course

This course contributes in a significant way to the accomplishment of the following program outcomes:

Program outcome	Emphasis in this course
(a) an ability to apply knowledge of computing and mathematics appropriate to the discipline	●
(b) an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution	◑
(c) an ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs	◑
(d) an ability to function effectively on teams to accomplish a common goal	
(e) an understanding of professional, ethical, legal, security and social issues and responsibilities	
(f) an ability to communicate effectively with a range of audiences	
(g) an ability to analyze the local and global impact of computing on individuals, organizations, and society	
(h) recognition of the need for and an ability to engage in continuing professional development	
(i) an ability to use current techniques, skills, and tools necessary for computing practice	○
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices	◑
(k) An ability to apply design and development principles in the construction of software systems of varying complexity	◑

Emphasis: ● High; ◑ Medium; ○ Low; Blank – Nothing Specific Expected

7. Brief list of topics to be covered

i. Analyzing the complexity of algorithms
ii. Data design, data structures, and data abstraction
iii. Stack and Queue abstract data types
iv. Lists, sorted and unsorted, abstract data types
v. Programming with recursion
vi. Trees
vii. Heaps and graphs
viii. Searching and Sorting