1. **Course number and name**
   CMP 340 – Design and Analysis of Algorithms

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

3. **Instructor's or course coordinator's name**
   Dr. Gerassimos Barlas

4. **Textbook, title, author, and year**
   A. Levitin. *Introduction to the Design & Analysis of Algorithms*, 3rd edition, Pearson, Addison Wesley, 2012.

   Other supplemental materials
   T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein C. *Introduction to Algorithms*, 2nd edition, MIT Press/McGraw-Hill, Cambridge, MA, 2001.

5. **Specific course information**
   a. Brief description of content of the course (catalog description)
   Covers algorithmic analysis; algorithmic strategies; advanced searching and sorting algorithms; hashing, graph and spanning trees algorithms; topological sort; pattern matching; numerical algorithms; matrix operations; complexity classes; approximation algorithms; and basic computability theory.

   b. Prerequisites or co-requisites
   Prerequisites: CMP 305/COE 311 (Data Structures & Algorithms), and NGN 111 (Introduction to Statistical Analysis) or STA 201 (Introduction to Statistics for Engineering and Natural Sciences).

   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. Describe the characteristics of the major complexity classes (P class, NP class)
   2. Use algorithm design methods, such as exhaustive search, divide-and-conquer and dynamic programming, to develop efficient algorithms.
   3. Use advanced searching techniques, such as hashing and 2-3 trees
   4. Show how time and space complexities can be traded-off (e.g. distribution counting sort, hashing)
   5. Analyze numerical computations algorithms (e.g. matrix multiplication)
   6. Apply efficient algorithms, such as Quicksearch, in string/pattern matching problems.
   7. Use fundamental graph algorithms, like traversal, shortest path and spanning tree in the solution of real-life 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 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. Introduction.
ii. Analysis, principles and notations.
iii. Analyzing non-recursive algorithms.
iv. Analyzing recursive algorithms.
v. Brute-force algorithms.
vi. Exhaustive search.
vii. Decrease-and-conquer.
viii. Divide-and-conquer algorithms.
ix. Transform-and-conquer techniques.
x. Space-time tradeoffs.
xi. Dynamic programming.
xii. Greedy algorithms.
xiii. P, NP and NP-complete problems.
xiv. Approximation algorithms.