1. **Course Number and Course Title:**

   CMP 340 – Design and Analysis of Algorithms

2. **Credit Hours:**

   3 – 0 – 3

3. **Prerequisites and/or Co-Requisites:**

   Prerequisites: CMP 305 and (STA 201 or STA 202 or NGN 111 or QBA 201)

4. **Name and Contact Information of Instructor:**

   Name: Dr. Salam Dhou

5. **Course Description (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.

6. **Textbook and other Supplemental Material:**

   Textbook:

   - A. Levitin, *Introduction to the Design & Analysis of Algorithms*, 3th ed., Addison-Wesley/Pearson, 2012.

   Other supplemental material:

   - T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms,* 3rd ed., MIT Press, 2009.

7. **Course Learning Outcomes:**

   Upon completion of the course, students will be able to:

   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.

8. **Teaching and Learning Methodologies:**

   Methods include lectures, problem-based learning, class discussions, and group work. Students learning is assessed via in-class quizzes, exams, homework, and team project.

9. **Course Topics and Schedule:**

Topic/Activity	**Weeks**
Introduction to algorithms	Week 1
Analysis, principles and asymptotic notations	Week 2

Analyzing non-recursive algorithms	Week 3
Analyzing recursive algorithms	Week 4
Brute-force algorithms: selection sort, bubble sort, brute-force string matching, and sequential search	Week 5
Decrease-and-conquer algorithms: insertion sort, topological sort, generating permutations and subsets, Russian Peasant multiplication, selection and partitioning algorithms	Week 6
Divide-and-conquer algorithms: numeric calculation algorithms (large integer multiplication, Strassen's matrix multiplication)	Week 7
Transform-and-conquer techniques: AVL trees, Heaps and Heapsort	Week 8
Space-time tradeoffs: counting sort, quicksearch and string/pattern matching	Week 9
Space-time tradeoffs: hashing, 2-3 trees, and B-trees	Week 10
Dynamic programming	Week 11
Greedy algorithms: minimum spanning tree and shortest path algorithm	Week 12
P, NP and NP-complete problems	Week 13
Approximation algorithms	Week 14
Revision and team project presentations	Week 15
Final Exam	Week 16