1. **Course number and name**
   CptS 350: Design and Analysis of Algorithms

2. **Credits and contact hours**
   3 credits, 3 lecture hours

3. **Instructor's or course coordinator's name**
   Zhe Dang

4. **Textbook, title, author, and year**
   T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. 2009. *Introduction to Algorithms* (3rd ed.).

5. **Specific course information**
   a. *Catalog description:* Analysis of data structures and algorithms; computational complexity and design of efficient data-handling procedures.
   b. *Prerequisites or corequisites:* CPT S 223 with a C or better or CPT S 233 with a C or better; CPT S 317 with a C or better; certified major in Computer Science, Computer Engineering, Electrical Engineering, or Software Engineering.

6. **Specific goals for the course**
   By the end of the course, students will be able to
   - Analyze complexities of algorithms using recurrence growth rate estimation (1b).
   - Design algorithms using basic algorithm design principles learned in the course to solve problems (1a,1c,1e).
   - Understand and use symbolic algorithms in manipulating large data structures (1d,1c,6a,7a,7f).

7. **Brief list of topics to be covered**
   - What is an algorithm? Fundamentals
   - Worst-case and average time complexities
   - Comparison-based sorting: lower complexity bound
   - Quick_Select: complexity analysis
   - MergeInsert: complexity analysis
   - Divide and conquer: Karatsuba algorithm and closest pair algorithm
   - Dynamic programming: LCS algorithm and a generalized LCS algorithm, applications in bioinformatics
   - Greedy algorithms: Huffman code and analysis
   - Amortized analysis: aggregate method, accounting method, potential method
   - Basic graph algorithms and analysis: DFS, BFS, topological sort, minimal spanning tree, shortest path
   - Advanced graph algorithms and applications: SCC, machines/programs as graphs, search over symbolic graphs
   - Number-theoretic algorithms: RSA and security protocols
   - NP-completeness, many-to-one reduction, SAT, 3SAT

- Automata-theoretic algorithms