CSCI 161 Computer Science II

1. **Course number and name:** CSCI 161 Computer Science II

2. **Credit and contact hours:** 4 credits, Lecture MWF 10:00 – 10:50am, Lab TTh 10:00 - 10:50am.
3. **Instructor's or course coordinator's name:** Sameer Abufardeh

4. **Textbook and other supplemental materials:** Data Structures and Algorithms in Java, Goodrich, Michael and Roberto Tamassia, 5th edition, John Wiley & Sons, Inc.

5. **Specific course information:**
   a. **Catalog description:** Advanced concepts in computer science including data structures, algorithm analysis, standard problems such as searching and sorting, and memory management issues.

   b. **Prerequisites:** CSCI 160 or equivalent
   c. **Required, elective, or selected elective:** Required for Computer Engineers.

6. **Specific goals for the course:**

   a. **Specific outcomes of instruction:**

      After completing this course, students should be able to:
      - Solve problems presented in a variety of formats.
      - Understand the basics of the software development lifecycle and the process of developing a small to medium size software application.
      - Use tools that support software development.
      - Understand and use key notions of object-oriented programming.
      - Understand and use fundamental data structures.
      - Understand and use techniques for analyzing the efficiency of algorithms and tradeoffs among algorithms.

   b. **Criterion 3 Student Outcomes addressed by this course:**

      (b) an ability to design and conduct experiments, as well as to analyze and interpret data.

      (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.

      (e) an ability to identify, formulate, and solve engineering problems.

      (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

7. **Brief list of topics to be covered:**
    - Software development lifecycles
    - Basic data types and their implementations
    - Abstract data type
    - The ways in which the lifetime of an object may be controlled
    - Use of recursion and iteration for solving problems
    - Some sorting algorithms
    - Efficient algorithms for identifying permutations and combinations
    - Design of algorithms
    - Inheritance and polymorphism
    - Abstract classes
    - Collections, iterators, and their appropriate uses
    - Classes of composite data structures