Course Name: Programming Language I

Course Code: CSE115

Credit Hours: 3

Faculty Information:

Name:

Designation:

Initials:

Email:

Office Hours:

Office Location:

Undergraduate Teaching Assistant: TBA

Course Description:

This course introduces basic methods for the design and analysis of efficient algorithms emphasizing methods useful in practice. Different algorithms for a given computational task are presented and their relative merits are evaluated based on performance measures. The following important computational problems will be discussed: sorting, searching, elements of divide-and-conquer, dynamic programming, greedy algorithms, advanced data structures, graph algorithms (shortest path, spanning trees, tree traversals), string matching, and NP-completeness.

Course Objectives:

-
  - Analyze the asymptotic performance of algorithms.
  - Write rigorous correctness proofs for algorithms.
  - Demonstrate familiarity with major algorithms and data structures.
  - Apply important algorithmic design paradigms and methods of analysis.
  - Synthesize efficient algorithms in common engineering design situations.

Course Outcomes (COs):

CO	CO Description
CO1	**analyze** time complexity of algorithms using asymptotic analysis.
CO2	**apply** different algorithm paradigms for a given problem

CO3	**explain** the major graph algorithms and their analyses to model engineering problems.

Lecture Plan:

Lecture	Topic
1	Introduction
2	Getting Started
3	Getting Started (Continues)
4	Growth of functions
5	Growth of functions (Continues)
6	Divide and Conquer Algorithms
7	Divide and Conquer Algorithms (Continues)
8	Divide and Conquer Algorithms (Continues)
9	Sorting
10	Sorting (Continues)
11	Sorting (Continues)
12	Sorting (Continues)
13	Greedy Algorithms
14	Greedy Algorithms (Continues)
15	Greedy Algorithms (Continues)

Lecture	Topic
16	**Mid-term Exam**
17	Dynamic Programming
18	Dynamic Programming (Continues)
19	Dynamic Programming (Continues)
20	Dynamic Programming (Continues)
21	Graph Algorithms
22	Graph Algorithms (Continues)
23	Graph Algorithms (Continues)
24	Graph Algorithms (Continues)
25	Graph Algorithms (Continues)
26	Graph Algorithms (Continues)
27	String Processing Algorithms
28	String Processing Algorithms (Continues)
29	Review

Weightage Distribution:

Assessment Tools	Weight

Class Performance	10%
Quizzes	25%
Project	15%
Mid-term Exam	20%
Final Exam	30%
Total	100%

Required Text Materials:

"Introduction to Algorithms", 4th Edition by Charles E. Leiserson, Clifford Stein, Ronald Rivest, and Thomas H. Cormen.

Reference Text and Materials:

"Algorithms", 4th Edition by Robert Sedgewick and Kevin.

Grading Policy:

http://www.northsouth.edu/academic/grading-policy.html (Links to an external site.)

Class Policies:

-
  - **Course Structure:** This course will be delivered offline, and course materials will be shared through the Learning Management System CANVAS.
  - **Quizzes and Exams:** There will be four quizzes. The best three quizzes will be counted for the final grading. There will be one midterm and one final exam.
  - **Project:** At the end of the semester students will complete a project. The detail will be announced after the mid-term exam.
  - **Late Work Policy:** Be sure to pay close attention to deadlines—there will be no make-up quizzes, exams, or late work accepted without a serious and compelling reason and faculty approval before the due date.
  - **Commit to Integrity:** As a student in this course (and at this university), you are expected to maintain high degrees of professionalism, commitment to active learning and participation in this class, and integrity in your behavior in and out of the classroom.

- **Academic Honesty Policy:** North South University system believes that academic honesty and integrity are fundamental to the mission of higher education. Students are responsible for the honest completion and representation of their work, for the appropriate citation of sources, and for respect for others' academic endeavors. Students who violate these standards must be confronted and must accept the consequences of their actions. Academic misconduct includes, but are not limited to:
  - cheating on a quiz or examination;
  - collaborating with others in work to be presented, contrary to the stated rules of the course;
  - submitting a paper or assignment as one's own work when a part or all of the paper or assignment is the work of another;
  - submitting, if contrary to the rules of a course, work previously presented in another course;
  - tampering with the laboratory experiment or computer program of another student;
  - knowingly and intentionally assisting another student in any of the above, including assistance in an arrangement whereby any work, classroom performance, examination, or other activity is submitted or performed by a person other than the student under whose name the work is submitted or performed.