
COURSE PREFIX/NO:	EGR 283
COURSE TITLE:	INTRODUCTION TO ALGORITHMIC DESIGN II
LEC HRS/WEEK:	3.0
LAB HRS/WEEK:	3.0
CREDIT HRS/SEMESTER:	4.0

[Distance Learning Attendance/VA Statement](#)
[Textbook Information](#)

COURSE DESCRIPTION

This course is a study of vigorous development of algorithms and computer programs, including elementary data structures.

COURSE COMPETENCIES

Upon successful completion of this course, the student should be competent to perform the following tasks:

Module 1: Introduction to Algorithms

Develop structured, modular algorithms.

Module 2: Object Oriented Language

Implement correct programs in an object-oriented language.

Module 3: Data Structures

Use and implement as classes data structures, such as sets, bags, sequences, stacks, queues, and binary trees.

Module 4: Simple Algorithms

Analyze the time and space complexity of simple algorithms.

Module 5: Complex Programs

Implement moderately complex programs written in an object-oriented language.

MINIMAL STANDARDS/PERFORMANCE OBJECTIVES:

Module 1: Introduction to Algorithms

Given a Java Integrated Development (IDE) system, the student will be able to write a program using structured, modular algorithms.

Module 2: Object Oriented Language

Given a Java Integrated Development (IDE) system, the student will be able to write a correct object-oriented program.

Module 3: Data Structures

Given a Java Integrated Development (IDE) system, the student will be able to write a program using data structures such as sets, bags, sequences, stacks, queues, and binary trees.

Module 4: Simple Algorithms

Given a simple Java program, the student will be able to analyze its time and space complexity.

Module 5: Complex Programs

Given a Java Integrated Development (IDE) system, the student will be able to write a moderately complex program.

COURSE REQUIREMENTS

Students are responsible for attaining competencies through completion of the following course requirements:

ATTENDANCE

Students should adhere to the attendance policy set forth in the York Technical College Handbook and Catalog. "Students who are absent from a class more than 10% of the hours assigned may be withdrawn." In case a student does miss a class, he/she is responsible for obtaining the material that was covered during the absence. If a student is aware that he/she will miss a class, then the student should notify the instructor at the earliest possible date.

If a student misses a test because of illness or urgent emergency, then he/she should do the following: Notify the instructor prior to the class period, or at the earliest possible date. At that time a new date for the make-up test will be scheduled. Students with unexcused absences when a test is given will be allowed to take a make-up test at the discretion of the instructor. The student has the responsibility to be sure that some arrangement is made with the instructor for taking a make-up test.

ACADEMIC INTEGRITY

Students are bound by the policies stated in the York Technical College Handbook and Catalog. Any student violating the policy will be subject to academic discipline.

ACADEMIC HONESTY

York Technical College adheres to the South Carolina TECH Students Code, approved by the State Board for Technical and Comprehensive Education on March 15, 1974 (revised last April 25, 1984). Copies of this code are available in the Library and from Student Services. "Any student caught cheating or involved in any other academic dishonesty will be given a grade of zero and will be subject to further disciplinary action." See the student handbook section "Student Life" subheading "Student Conduct" for further details.

PARTICIPATION IN CLASS DISCUSSIONS, PROBLEM-SOLVING

COMPLETING ASSIGNED HOME WORK, COMPUTER PROGRAMS AND WRITTEN TESTS

EVALUATION STRATEGIES/GRADING

The following scale will be the standard:

Grade Score

A 90 - 100

B 80 - 89

C 70 - 79

D 60 - 69

F below 60

Each module will count equally toward the final grade. Each module will have at least one computer program or test.

Final grades will be determined as follows:

Major Tests	60 %
Laboratory	15 %
Instructor Options	15 %
Affective Skills	<u>10 %</u>
	100 %

The instructor options will be printed on handouts with the grading percent and given to the students during the first week of class. These options may include homework, spot quizzes or written reports.

The affective skills, which count as 10% of the total grade, are evaluated by the following:

1. Attendance at 95% or more of the classes and labs. (25% of 10%)
2. Dependability (No tardiness). (25% of 10%)
3. Acceptable student conduct as defined in the College Catalog. (50% of 10%)

Students are expected to "conduct themselves with dignity and to maintain high standards of responsible citizenship." (York Technical College Catalog, Student Life Section. "Student Conduct") Points will be deducted for disruptive behavior which violates the principles of acceptable conduct.

ENTRY-LEVEL SKILLS: None

PREREQUISITES: EGR 281

CO-REQUISITES: None

METHODS OF INSTRUCTION

This course consists of 3 hours of class periods and 3 hours of laboratory time each week. The class instruction includes lectures, discussions, problem-solving sessions, and tests. The lectures are administered using the whiteboard, multimedia devices, computer simulations, and video tapes. The discussions consist of student-student and student-instructor dialogue. The laboratory exercises follow and complement the class lectures. The students are given instructions before the laboratory begins.

Disabilities Statement: Any student who feels s/he may need an accommodation based on the impact of a disability should contact the Special Resources Offices (SR) at 803-327-8007 in the 300 area of Student Services. The SRO coordinates reasonable accommodations for students with documented disabilities.

TOPIC/CONTENT OUTLINE

Module 1: Introduction to Algorithms
The phases of software development
Abstract data types and Java classes

Module 2: Object Oriented Language
Collection classes
Linked Lists

Module 3: Data Structures
Object Types
Stacks

Module 4: Simple Algorithms
Queues
Recursion

Module 5: Complex Programs
Binary Search and Analysis of Algorithms
Trees