

J. Sargeant Reynolds Community College
Course Content Summary

Course Prefix and Number: EGR 125 Credits: 4

Course Title: Introduction to Computer Programming for Engineers

Course Description :

Introduces problem solving and implementation of computer software solutions using a high-level programming language in a structured environment. Includes concepts and practice of algorithm design, language syntax, control structures, arrays, and introduction to object-oriented programming. Covers engineering applications, such as mathematical modeling, file input and output, and basic numerical methods. The assignments in this course require mathematical problem-solving skills, algebraic modeling, and functions, and use of variables. Lecture 4 hours, Total 4 hours per week. 4 credits. Credit may not be earned for both CSC 221 and EGR 125.

General Course Purpose:

EGR 125 is an introductory course in computer programming for engineers. It serves as the standard sequence of minimal programming content for all engineering majors and the first course in a programming sequence for electrical and electrical computer engineers. It is similar to the first course in computer science (CSC 221) with additional engineering programming applications included. Students may not earn credit for both. This course introduces computer software-based problem solving and implementation of solutions in a high-level programming language.

Course Prerequisites/Corequisites:

Prerequisites: MTH TJ 0 Tc 0 Tw 19.054 0 Td ()Tj EMC /LBody <</MCID 38 >>BDC -20.685 -1.141 Td [(•)2 (

- Analyzing and resolving run-time errors.

Problem analysis and algorithmic modeling

- List and apply the steps involved in problem solving through algorithmic modeling.
- Describe activities related to program modeling and design including algorithm development.
- Solve problems using techniques such as pseudocode, flowcharts, and model development.
- Verify algorithms and identify errors.
- Distinguish between procedural techniques and object-oriented techniques.
- Write programs using good programming practices.

Use of data

- Compare and contrast data types.
- Describe the use of variables.
- Build expressions using variables, literal data, and operators, correctly using rules of operator precedence.

Decision structures

-

Classes and Introduction to Libraries

- Describe information hiding and encapsulation.
- Describe the concept of class and object of a class.
- Use language classes from the standard library to develop programs.

Engineering Applications

- File input and output.
- •Introduction to mathematical modeling
- •Numerical methods, including root finding and numerical integration

Major Topics to be Included:

- Basic concepts of computer systems
- Processing Code
- Problem analysis and algorithmic modeling
- Use of data
- Decision structures
- Repetition structures
- Programming with Procedures
- Collections
- Memory Allocation
- Classes and Introduction to Libraries
- Engineering Applications

Effective Date/Updated: August 1, 2023