TAFT COLLEGE
West Kern Community College District

Prepared by: P. Blake
Reviewed by: G. Golling
Reviewed by: S. Aunai
Date prepared: October 2013
C&GE approved: November 13, 2013
Board approved: December 11, 2013
Text Updated: October 24, 2016

Engineering (ENGR) 1540 Introduction to Programming Concepts and Methodologies for Engineers with Lab (3 Units) CSU:UC

Prerequisite: Successful completion of MATH 1540 with a grade of "C" or better

Prerequisite knowledge/skills: Before entering the course the student should be able to:

1. understand the concepts of functions and inverse functions, and use functional and inverse functional notation. Apply this knowledge to linear and quadratic functions as an introduction to the use of these concepts,
2. understand rational, polynomial, exponential, logarithmic, trigonometric and inverse trigonometric functions, and identities involving these functions,
3. graph any function covered in the course,
4. solve linear systems of equations by standard elementary techniques and by using the Gaussian reduction method and/or Cramer's rule,
5. use mathematical induction to solve sequence and series problems, including the development of the binomial theorem, and
6. apply any of the mathematical methods covered in this course to the solution of appropriate practical problems chosen from a variety of fields.

Advisory: basic knowledge of computer usage and eligibility for English 1500 strongly recommended

Total Hours: 32 hours lecture; 48 hours lab (80 hours total)

Catalog Description: This course covers the fundamental concepts of procedure-oriented programming, associated abstraction mechanisms and design processes, the interface of software with the physical world (e.g., the use of sensors), and the application of numerical techniques. This course uses the C++ programming language.

Type of Class/Course: Transfer Degree Credit

Text: Gaddis, Walters and Muganda. *Starting out with C++ Early Objects*. 9th ed. Pearson, 2017.

Course Objectives:

By the end of the course a successful student will be able to:

1. Describe the basics of the architecture of a computer and its components,
2. Describe the principles of structured programming,
3. Describe, design, implement, and test structured programs using currently accepted methodology, and in particular to be able to do so for programs that control or otherwise interfaces with hardware by means of software,

4. Explain what an algorithm is and its importance in computer programming, and
5. Apply numerical techniques to analyze and solve engineering-related problems.

Course Scope and Content (Lecture):

Unit I — Software Lifecycle
- A. Architecture Design
- B. Code Development
- C. Development Styles
- D. Documentation
- E. Testing
- F. Maintenance

Unit II — Development Approaches
- A. Procedural
- B. Object Oriented

Unit III — Program Design Tools & Programming Environments
- A. Languages
- B. Compilers, Linkers, Debuggers, Development Kits
- C. Environments

Unit IV — Coding Conventions
- A. Styles
- B. Business vs. Technical
- C. Configuration Management

Unit V — Data Types
- A. Variables
- B. Constants
- C. Expressions
- D. Functions
- E. Sequential Processing

Unit VI — Arrays
- A. Declaring and allocating arrays
- B. Multiple-subscripted arrays

Unit VII — Control structures
- A. Selective structures: if then and switch
- B. Repetitive structures: loops
- C. Functional and procedural abstraction

Unit VIII — Interfacing with the Physical World
- A. User Interfaces
- B. Sensors
- C. Networks
- D. Internet

Unit IX — Algorithms
- A. Sorting
- B. Searching

C. Random Generation
D. Computational Calculations

Unit X Parameter Passing
A. Value
B. Reference

Unit XI Testing
A. Principles
B. Environment
C. Data Sets
D. Procedures

Unit XII Data I/O (Input/Output)
A. Data Files
B. Sensors
C. User Interfaces

Course Scope and Content (Lab)

Unit I Introduction
A. Basic software development tools
B. Basic hardware interface tools
C. Lab Procedures
D. Development Toolkit

Unit II Software Coding
A. Functions
B. Arrays
C. Variables
D. Loop Controls

Unit III Software Processes
A. Coding
B. Compiling
C. Linking
D. Debugging

Unit IV Data Manipulation
A. Sorting
B. Parsing
C. Searching

Unit V Data I/O
A. File input and File Output
B. User Interface

Unit VI Hardware interfaces
A. Input from User Interfaces
B. Manipulation of Hardware

C. Sensors and Switches

Unit VII Test Environment
A. Quality Assurance
B. Functional Testing
C. Test Sets
D. Design Evaluations

Learning Activities Required Outside of Class:

The students in this class will spend a minimum of 4 hours per week outside of the regular class time doing the following:

1. Studying assigned text, handout materials and class notes
2. Reviewing and preparing for quizzes, midterm and final exams
3. Completing individual homework assignments following coding guidelines and proper documentation.

Methods of Instruction:

1. Lecture, demonstrations and discussions
2. Individual homework and lab assignments with emphasis on proper coding and development techniques and problem solving.

Methods of Evaluation:

1. Quizzes
2. Exams
3. Participation
4. Individual assignments and group assignments
5. Design Project and Presentation

Laboratory Category: Extensive Laboratory

Pre delivery criteria: All of the following criteria are met by this lab.
1. Curriculum development for each lab.
2. Published schedule of individual laboratory activities.
3. Published laboratory activity objectives.
4. Published methods of evaluation.
5. Supervision of equipment maintenance, laboratory setup, and acquisition of lab materials and supplies.

During laboratory activity of the laboratory: All of the following criteria are met by this lab.
1. Instructor is physically present in lab when students are performing lab activities.
2. Instructor is responsible for active facilitation of laboratory learning.
3. Instructor is responsible for active delivery of curriculum.
4. Instructor is required for safety and mentoring of lab activities.

5. Instructor is responsible for presentation of significant evaluation.

Post laboratory activity of the laboratory: All of the following criteria are met by this lab.

1. Instructor is responsible for personal evaluation of significant student outcomes (lab exercises, exams, practicals, notebooks, portfolios, etc.) that become a component of the student grade that cover the majority of lab exercises performed during the course.
2. Instructor is responsible for supervision of laboratory clean up of equipment and materials.

Supplemental Data:

| | |
|---|---|
| TOP Code: | 090100: Engineering, General (requires |
| SAM Priority Code: | E: Non-Occupational |
| Distance Education: | Not Applicable |
| Funding Agency: | Y: Not Applicable(funds not used) |
| Program Status: | 1: Program Applicable |
| Noncredit Category: | Y: Not Applicable, Credit Course |
| Special Class Status: | N: Course is not a special class |
| Basic Skills Status: | N: Course is not a basic skills course |
| Prior to College Level: | Y: Not applicable |
| Cooperative Work Experience: | N: Is not part of a cooperative work experience education program |
| Eligible for Credit by Exam: | E: Credit By Exam |
| Eligible for Pass/No Pass: | NO |
| Taft College General Education: | NONE |