TAFTCOLLEGE

Reviewed by: P. Blake
Reviewed by: M. Mayfield
Date Reviewed: Spring 2022
Textbook update: Spring 2022
C & GE Update: April 21, 2022
Board approved: May 11, 2022
Semester effective:

Engineering (ENGR) 1540 Introduction to Programming Concepts and Methodologies for Engineers with Lab (4 Units) CSU: UC

Prerequisite: Successful completion of MATH 1540 Precalculus Mathematics with a grade of 'C' or better

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

1. Graph functions and relations in rectangular coordinates and polar coordinates;
2. Synthesize results from the graphs and/or equations of functions and relations;
3. Apply transformations to the graphs of functions and relations;
4. Recognize the relationship between functions and their inverses graphically and algebraically; 5. Solve and apply equations including rational, linear, polynomial, exponential, absolute value, radical, and logarithmic, and solve linear, nonlinear, and absolute value inequalities;
6. Solve systems of equations and inequalities;
7. Apply functions to model real world applications;
8. Identify special triangles and their related angle and side measures;
9. Evaluate the trigonometric function of an angle given in degree and radian measure;
10. Manipulate and simplify a trigonometric expression;
11. Solve trigonometric equations, triangles, and applications;
12. Graph the basic trigonometric functions and apply changes in period, phase and amplitude to generate new graphs; and
13. Prove trigonometric identities

Advisory: basic knowledge of computer usage and eligibility for English 1500 or 1501 strongly recommended

Hours and Units Calculations:
48 hours lecture. 96 Outside of class hours. 48 hours lab. (192 Total Student Learning Hours) 4 Units

Catalog Description: This course introduces the fundamental concepts of procedural programming and object oriented programming. Topics include: data types, control structures, functions, arrays, I/O, pointers, dynamic memory allocation and features object-oriented programming. The mechanisms of compiling, linking, running, debugging and testing are covered, binding, visibility, scoping, and lifetime management are also included. Ethical Issues and a historical perspective in context of computer science and engineering is given. The interface of software with the physical world (e.g., the use of sensors), and the application of numerical techniques is also covered. This course uses the C++ programming language. C-ID ENGR 120

Type of Class/Course: Transfer Degree Credit

Text: Gaddis, Tony, et al. *Starting Out with C++: early Objects*. 10th ed., Pearson, 2019.
Additional materials:

TAFTCOLLEGE

Lab Manual

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
G. Lifecycle Management

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- dimensional arrays

Unit VII Control structures

A. Selective structures: if else and switch
B. Repetitive structures: loops

C. Functional and procedural abstraction

Unit VIII Interfacing with the Physical World
A. User Interfaces
B. Sensors

Unit IX Algorithms
A. Sorting
B. Searching
C. Random Generation
D. Computational Calculations

Unit X Functions and Parameter Passing
A. Value
B. Reference
C. Void
D. Scope
E. Visibility
F. Binding

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

Unit XIII Object Oriented Programming
A. Abstraction
B. Classes and Objects
C. Member Functions
D. Constructors and Destructors
E. Passing Objects to Functions
F. Private vs. Public Access

Course Scope and Content (Laboratory)

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 6 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
Discipline	Engineering or Computer Science