ENGR125 : Programming Concepts and Methodologies for Engineers

General Information

Author:	• Christopher Herwerth
Course Code (CB01) :	ENGR125
Course Title (CB02) :	Programming Concepts and Methodologies for Engineers
Department:	ENGR
Proposal Start:	Fall 2025
TOP Code (CB03) :	(0901.00) Engineering, General (requires Calculus) (Transfer)
CIP Code:	(14.0102) Pre-Engineering.
SAM Code (CB09) :	Non-Occupational
Distance Education Approved:	No
Will this course be taught asynchronously?:	No
Course Control Number (CB00) :	CCC000603903
Curriculum Committee Approval Date:	04/09/2025
Board of Trustees Approval Date:	Pending
Last Cyclical Review Date:	02/01/2020
Course Description and Course Note:	ENGR 125 teaches the application of structured programming concepts for engineering problems. Topics include practical mechanics, electronics, robotics, as well as the design, reduction and analysis of experimental data using high level languages. Required for electrical and computer engineering majors, students learn to program and control engineering devices and sensors that interface with the physical world. The interaction between computer programming and the physical world is a major focus of the course. Note: This course may not be used to meet the requirements of any computer science program.
Justification:	Content Change
Academic Career:	• Credit
Mode of Delivery:	No value
Author:	No value
Course Family:	No value

Academic Senate Discipline

Primary Discipline:	• Engineering
Alternate Discipline:	No value
Alternate Discipline:	No value

Course Development

Basic Skill Status (CB08)

Course is not a basic skills course.

☐ Allow Students to Gain Credit by Exam/Challenge

Course Special Class Status (CB13)

Course is not a special class.

Pre-Collegiate Level (CB21)

Not applicable.

Grading Basis

- Grade Only

Course Support Course Status (CB26)

Course is not a support course

General Education and C-ID

General Education Status (CB25)

Not Applicable

Transferability

Transferable to both UC and CSU

Transferability Status

Approved

C-ID	Area	Status	Approval Date	Comparable Course
ENGR	Engineering	Approved	09/03/2019	ENGR 120 - Introduction to Programming Concepts and Methodologies for Engineers

Units and Hours

Summary

Minimum Credit Units (CB07)	4
Maximum Credit Units (CB06)	4
Total Course In-Class (Contact) Hours	108
Total Course Out-of-Class Hours	108
Total Student Learning Hours	216

Credit / Non-Credit Options

Course Type (CB04)

Credit - Degree Applicable

Noncredit Course Category (CB22)

Credit Course.

Noncredit Special Characteristics

No Value

Course Classification Code (CB11)

Credit Course.

☐ Variable Credit Course

Funding Agency Category (CB23)

Not Applicable.

☐ Cooperative Work Experience Education Status (CB10)

Weekly Student Hours

Course Student Hours

	In Class	Out of Class
Lecture Hours	3	6
Laboratory Hours	3	0
Studio Hours	0	0

Course Duration (Weeks)	18
Hours per unit divisor	0
Course In-Class (Contact) Hours	
Lecture	54
Laboratory	54
Studio	0
Total	108
Course Out-of-Class Hours	
Lecture	108
Laboratory	0
Studio	0
Total	108

Time Commitment Notes for Students

No value

Units and Hours - Weekly Specialty Hours

Activity Name	Type	In Class	Out of Class
No Value	No Value	No Value	No Value

Prerequisites, Corequisites, Recommended Corequisites, and Recommended Preparation

Prerequisite

MATH103E - Calculus & Analytic Geometry I (in-development)

Objectives

- use the derivative for rate of change problems;
- use differentiation to solve applications such as related rate problems and optimization problems;
- use implicit differentiation with applications, including in differentiation of inverse functions;
- find derivatives of transcendental functions: trigonometric, exponential, logarithmic, and others;
- determine relative and absolute maximum and minimum points of functions and points of inflection;
- graph functions using the methods of calculus;
- use the Mean Value Theorem;
- evaluate a definite integral as a limit of Riemann sums;
- apply integration to find areas, apply properties of integrals;
- evaluate integrals using the Fundamental Theorem of Calculus;

AND

Advisory

ESL151 - Reading And Composition V

Objectives

- Read and critically analyze various academic readings.
- Summarize readings.
- Compose a 500 to 550-word essay which: summarizes and cites appropriately a reading passage; includes a clear thesis statement; uses evidence to support the thesis; shows clear organization into an introduction, body, and conclusion.
- Revise writing to eliminate errors in syntax, and grammatical constructions.
- Employ basic library research techniques.

Entry Standards

Entry Standards	Description
basic knowledge of computer usage.	No Value

Course Limitations

Cross Listed or Equivalent Course	Description
No value	No value

Requisite Validation

Upload Statistical Validation and/or other documents (if necessary)

No Value

Specifications

Methods of Instruction

Methods of Instruction	Lecture
Methods of Instruction	Laboratory
Methods of Instruction	Discussion
Methods of Instruction	Multimedia

Methods of Instruction	Collaborative Learning
Methods of Instruction	Demonstrations

Out of Class Assignments

homework (e.g. write an algorithm that solves a set of linear equations)
essay [e.g. summarize the activities of the Association for Computing Machinery (ACM), including industry standards]
project (e.g. use a microcontroller to design a digital thermometer)

Methods of Evaluation	Rationale
Exam/Quiz/Test	quizzes
Exam/Quiz/Test	laboratory assignments (e.g. program a microcontroller to use various sensors)
Exam/Quiz/Test	project (e.g. programming a microcontroller to collect air pollution data using a particulate matter sensor)
Exam/Quiz/Test	exams (e.g. midterm)
Exam/Quiz/Test	final exam

Textbook Rationale

No Value

Textbooks

Author	Title	Publisher	Date	ISBN
Walter Savitch	Problem Solving with C++	Pearson	2018	9780134448282

Other Instructional Materials (i.e. OER, handouts)

No Value

Learning Outcomes

Course Objectives

Demonstrate different forms of binding, visibility, scoping, and lifetime management;

integrate software and hardware components in order to respond to physical phenomena and manipulate physical devices and objects;

analyze and explain simple programs involving fundamental programming constructs;

modify and expand short programs that use standard conditional and iterative control structures and functions;

create algorithms for solving simple problems;

apply pseudocode or a programming language to implement, test, and debug algorithms for solving simple problems;

identify and describe the properties of a variable such as its associated address, value, scope, persistence, and size;

apply software developed by others, modify the software, and validate the quality of the modifications;

demonstrate the interaction between software and the physical world;

describe simple software quality assurance (QA) procedures;

demonstrate systematic QA procedures to evaluate design and implementation quality;

demonstrate awareness of industry standards for quality assurance and software life cycle such as ISO 9000 and IEEE.

SLOs

design, implement, test, and debug a program that uses fundamental programming constructs such as basic computation, input-output, conditional and iterative structures, and the definition of functions

Expected Outcome Performance: 70.0

ILOs Core ILOs	Analyze and solve problems using critical, logical, and creative thinking; ask questions, pursue a line of inquiry, and derive conclusions; cultivate creativity that leads to innovative ideas.
	Demonstrate depth of knowledge in a course, discipline, or vocation by applying practical knowledge, skills, abilities, theories, or methodologies to solve unique problems.
ENGR Civil Engineering	Apply knowledge of mathematics, science and engineering; identify, form and solve engineering problems
	Demonstrate introductory skills using modern engineering tools necessary for engineering practice.
ENGR Engineering Technology - CAD & Design Drafting	Demonstrate techniques to accomplish drawings and 3D models utilizing different various computer aided design (CAD) software
ENGR Engineering Entrepreneurship Skill Award	Learn hands-on skills and problem solving techniques for businesses related to engineering design, installation, manufacturing, testing, technical sales, maintenance, and other such topics in engineering technology.
	Learn the engineering design process and how technical products are made, assembled, and integrated into complex systems.
ENGR Mechanical Engineering - A.S. Degree Major	analyze engineering problems and make appropriate decisions with the supervision of a licensed engineer;
	design a system, component, or process with supervision of a licensed engineering to meet desired needs.
	use science and mathematical skills required for occupational needs;
ENGR Electrical Engineering A.S. Degree Major	analyze engineering problems and make appropriate decisions with the supervision of a licensed engineer;
	design a system, component, or process with supervision of a licensed engineer to meet desired needs.
	use science and mathematical skills required for occupational needs;
ENGR Computer Engineering AS	use science and mathematical skills required for occupational needs;

Summarize the evolution of programming languages and show how the classification of programming features have created the programming paradigms that are used today

Expected Outcome Performance: 70.0

ILOs Core ILOs	Demonstrate depth of knowledge in a course, discipline, or vocation by applying practical knowledge, skills, abilities, theories, or methodologies to solve unique problems.
ENGR Civil Engineering	Demonstrate introductory skills using modern engineering tools necessary for engineering practice.
ENGR Engineering Technology - CAD & Design Drafting	Discuss how the design process and design/drawing techniques are used with other engineering processes to create a finished product.
ENGR Engineering Entrepreneurship Skill Award	Learn the engineering design process and how technical products are made, assembled, and integrated into complex systems.
ENGR Electrical Engineering A.S. Degree Major	demonstrate appropriate technical written, verbal, drawing, and communication skills;
ENGR Mechanical Engineering - A.S. Degree Major	demonstrate appropriate technical written, verbal, drawing, and communication skills;
ENGR Computer Engineering AS	demonstrate appropriate technical written, verbal, drawing, and communication skills;

create programs using basic development tools and integrated development environments that allows for the interface with the physical world by using a microcontroller to record and display data or control a device Expected Outcome Performance: 70.0

ILOs Core ILOs	Analyze and solve problems using critical, logical, and creative thinking; ask questions, pursue a line of inquiry, and derive conclusions; cultivate creativity that leads to innovative ideas.
	Demonstrate depth of knowledge in a course, discipline, or vocation by applying practical knowledge, skills, abilities, theories, or methodologies to solve unique problems.
ENGR Civil Engineering	Apply knowledge of mathematics, science and engineering; identify, form and solve engineering problems
	Demonstrate introductory skills using modern engineering tools necessary for engineering practice.
ENGR Engineering Technology - CAD & Design Drafting	Demonstrate techniques to accomplish drawings and 3D models utilizing different various computer aided design (CAD) software
ENGR Engineering Entrepreneurship Skill Award	Learn hands-on skills and problem solving techniques for businesses related to engineering design, installation, manufacturing, testing, technical sales, maintenance, and other such topics in engineering technology.
	Learn the engineering design process and how technical products are made, assembled, and integrated into complex systems.
ENGR Mechanical Engineering - A.S. Degree Major	analyze engineering problems and make appropriate decisions with the supervision of a licensed engineer;
ENGR Computer Engineering AS	analyze engineering problems and make appropriate decisions with the supervision of a licensed engineer;
ENGR Electrical Engineering A.S. Degree Major	analyze engineering problems and make appropriate decisions with the supervision of a licensed engineer;
	use science and mathematical skills required for occupational needs;

Additional SLO Information

Does this proposal include revisions that might improve student attainment of course learning outcomes?

No Value

Is this proposal submitted in response to learning outcomes assessment data?

No Value

If yes was selected in either of the above questions for learning outcomes, explain and attach evidence of discussions about learning outcomes.

No Value

SLO Evidence

No Value

Course Content

Lecture Content

Introduction (2 hours)

- History of computer engineering
- Definitions and interdisciplinary engineering
- Review of basic computer usage
- Engineering design process applications in computer and electrical engineering

Programming Fundamentals (4 hours)

- Basic computer science concepts
- Syntax and semantics in high level languages
- Variables, types, expressions, and assignment
- Data types
- Simple input-output
- Simple decision-making statements
- Conditional and iterative control structures
- Functions and parameter passing
- Structured decomposition

Control Structures (4 hours)

- Top-down methodology
- Algorithms and stepwise refinement
- Conditional control structures
- Choosing the best iterative constructs for a programming task
- Iterative control structures
- If, if/else and switch selection structures
- While, do/while repetition structures

Functions (4 hours)

- Functions and parameter passing
- Mechanics of parameter passing
- Structured decomposition
- Techniques of functional decomposition to break a program into smaller pieces
- Simulation techniques using random number generation
- Visibility of identifiers and limitations
- Functions that call themselves
- Recursion

Arrays (4 hours)

- Array data structure
- Declare and initialize arrays
- Passing arrays to functions
- Array elements and indices
- Basic sorting techniques
- Declare and manipulate multiple subscript arrays
- Limits of data in microcontrollers

Algorithms and Problem Solving (4 hours)

- Engineering problem solving strategies and methods
- Concept and properties of algorithms
- Identifying effective properties of strong algorithms
- Importance of algorithms in the engineering design and problem-solving processes
- Algorithms as problem solving and organization tools
- Implementation strategies using algorithms
- Debugging skills and strategies
- Communicating effective algorithm and debugging techniques

Programming Languages Used in Engineering (4 hours)

- History and overview of programming language development in the context of engineering needs
- Survey of programming paradigms
- Procedural languages
- Object-oriented languages
- Distinguishing characteristics of programming paradigms

Declaration and Types (4 hours)

- Conception of types as a set of values with a set of operations
- Declaration models
- Blinding
- Visibility

- Scope
- Lifetime
- Value of declaration models with respect to programming-in-the-large
- Type checking
- Type incompatibility
- Importance of types and type-checking in providing abstraction and safety

Interface with the Physical (8 hours)
- World Software development tools
- Basic hardware interface development tools
- Modifying software
- Demonstration of designs and implementations
- Implementing software that manipulates or responds to physical phenomena

Interface with the Physical World (continued) (8 hours)
- Embedded programming in the microcontroller
- Using microcontrollers that are preprogrammed
- Programming for sensors
- Programming for servo motors
- Graphical user interface
- Circuit fundamentals and building circuits for sensors and actuators
- Different types of commercially available microcontroller platforms such as Arduino, Raspberry Pi, National Instruments and LABVIEW
- Using other software tools such as MATLAB
- Basics of signal processing such as amplification and noise reduction to obtain and use sensor data
- Robotics and variations on microcontroller uses
- Automatic and autonomous functioning of microcontrollers embedded in everyday devices

Pointers and Strings (4 hours)
- Using pointers to pass arguments to functions by reference
- Relationships between pointers, arrays, and strings
- Declare and use arrays of strings

File Processing (4 hours)
- Create, read, write, and update files
- Sequential-access file processing
- Random-access file processing
- Formatted data vs. raw data file processing

Total 54 hours

Laboratory/Studio Content

Introduction (2 hours)
- History of computer engineering
- Definitions and interdisciplinary engineering
- Review of basic computer usage
- Engineering design process applications in computer and electrical engineering

Programming Fundamentals (4 hours)
- Basic computer science concepts
- Syntax and semantics in high level languages
- Variables, types, expressions, and assignment
- Data types
- Simple input-output
- Simple decision-making statements
- Conditional and iterative control structures
- Functions and parameter passing
- Structured decomposition

Control Structures (4 hours)
- Top-down methodology
- Algorithms and stepwise refinement
- Conditional control structures
- Choosing the best iterative constructs for a programming task
- Iterative control structures
- If, if/else and switch selection structures

- While, do/while repetition structures

Functions (4 hours)
- Functions and parameter passing
- Mechanics of parameter passing
- Structured decomposition
- Techniques of functional decomposition to break a program into smaller pieces
- Simulation techniques using random number generation
- Visibility of identifiers and limitations
- Functions that call themselves
- Recursion

Arrays (4 hours)
- Array data structure
- Declare and initialize arrays
- Passing arrays to functions
- Array elements and indices
- Basic sorting techniques
- Declare and manipulate multiple subscript arrays
- Limits of data in microcontrollers

Algorithms and Problem Solving (4 hours)
- Engineering problem solving strategies and methods
- Concept and properties of algorithms
- Identifying effective properties of strong algorithms
- Importance of algorithms in the engineering design and problem-solving processes
- Algorithms as problem solving and organization tools
- Implementation strategies using algorithms
- Debugging skills and strategies
- Communicating effective algorithm and debugging techniques

Programming Languages Used in Engineering (4 hours)
- History and overview of programming language development in the context of engineering needs
- Survey of programming paradigms
- Procedural languages
- Object-oriented languages
- Distinguishing characteristics of programming paradigms

Declaration and Types (4 hours)
- Conception of types as a set of values with a set of operations
- Declaration models
- Blinding
- Visibility
- Scope
- Lifetime
- Value of declaration models with respect to programming-in-the-large
- Type checking
- Type incompatibility
- Importance of types and type-checking in providing abstraction and safety

Interface with the Physical (8 hours)
- World Software development tools
- Basic hardware interface development tools
- Modifying software
- Demonstration of designs and implementations
- Implementing software that manipulates or responds to physical phenomena

Interface with the Physical World (continued) (8 hours)
- Embedded programming in the microcontroller
- Using microcontrollers that are preprogrammed
- Programming for sensors
- Programming for servo motors
- Graphical user interface
- Circuit fundamentals and building circuits for sensors and actuators
- Different types of commercially available microcontroller platforms such as Arduino, Raspberry Pi, National Instruments and LABVIEW
- Using other software tools such as MATLAB
- Basics of signal processing such as amplification and noise reduction to obtain and use sensor data

- Robotics and variations on microcontroller uses
- Automatic and autonomous functioning of microcontrollers embedded in everyday devices

Pointers and Strings (4 hours)
- Using pointers to pass arguments to functions by reference
- Relationships between pointers, arrays, and strings
- Declare and use arrays of strings

File Processing (4 hours)
- Create, read, write, and update files
- Sequential-access file processing
- Random-access file processing
- Formatted data vs. raw data file processing

Total 54 hours

Additional Information

Repeatability

Not Repeatable

Justification (if repeatable was chosen above)

No Value

Is it possible this course will have a material fee?

No Value

I have contacted my library liaison (https://campusguides.glendale.edu/faculty/liaisons):

No Value

What term(s) will this course be offered?

No Value

Will any additional resources be needed for this course? (Click all that apply)

No Value

If additional resources are needed, add a brief description and cost in the box provided.

No Value