PARKLAND COLLEGE

Course Information Form (CIF)

Course Code: CSC 127

Course Title: Introduction to Computing with Engineering Applications

Department: Business/Computer Science and Technologies
Effective Date: Summer 2026
PCS Code: 1.1 - Baccalaureate/Transfer
CIP Code: 11.0202
Repeatability: 0

Credit Hours

Catalog Notation: 2-2-3
Credit Hour Distribution:

Lecture: 2	Lab: 2	Clinical: 0	**Total: 3**

General Course Information

Catalog Description

Fundamental principles, concepts, and methods of computing with emphasis on applications in the physical sciences and engineering. Basic problem solving and programming techniques, fundamental algorithms and data structures, use of computers in solving engineering problems and numerical analysis.

General Course Objectives

This is an introductory computer science course intended primarily for physical science and engineering students. It is not intended for computer science majors. It emphasizes fundamental concepts, problem solving, and algorithms of use in engineering and the physical sciences.

Minimum Placement Levels

English	Reading	Math
None	None	Credit in MAT 128 with a grade of C or higher, or placement

Prerequisites

None

Methods of Evaluation

One midterm examination which contains programming problems, a comprehensive final exam, 6-10 graded projects, and lab exercises.

Instructional Materials and Additional Supplies

Online textbooks:

- https://greenteapress.com/thinkpython2/html/index.html
- https://www.halvorsen.blog/
- https://en.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial

Course Content

General Learning Outcomes (GLOs)

- Reasoning and Inquiry: Students will demonstrate the ability to solve problems using deductive reasoning and logic, quantitative reasoning, or the scientific method.
- Technology: Students will demonstrate the ability to evaluate, select, and appropriately use current and emerging tools.

Course Segments and Student Learning Outcomes

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Basics of Computer Programming	1. Identify the components of an IDE. 2. Identify the parts of a minimal computer program and demonstrate the ability to make small modifications to it. 3. Outline the history of programming languages and identify different languages. 4. List intrinsic data types and apply them to real world data. 5. Reproduce the conversions between decimal, binary, and hexadecimal number systems.	2	2	0
Expressions and Functions in Programming	1. Identify and explain mathematic operators. 2. Recognize and reproduce variables. 3. Demonstrate expressions built of variables and operators. 4. Identify parts of a function: input parameters, processing, and output values. 5. Construct functions to solve specific problems.	4	4	0
Decisions	1. Identify and describe the comparison operators. 2. Identify and describe the logical operators AND, OR, and NOT. 3. Predict the flow of control around decision control structures. 4. Apply decision control structures to real world problems.	2	2	0
Looping and Repetition	1. Predict the flow of control around repetitive control structures. 2. Apply repetitive control structures to real world problems. 3. Apply multiple different control structures to solve real world problems.	2	2	0
Program Design	1. Define top-down design principles and concepts. 2. Break down a significant problem using top-down design. 3. Describe local, global, and other scopes of data in larger programs.	2	2	0
String Manipulation	1. Manipulate string objects. 2. Manipulate ASCII text files.	2	2	0
Arrays	1. Describe how arrays are physically stored in memory. 2. Apply looping control structures to manipulate arrays. 3. Describe sorting algorithms.	2	2	0
Vectors and Matrices	1. Identify vector methods. 2. Apply vector methods to solve real world problems. 3. Apply vectors and arrays to manipulate 2D matrices.	2	2	0

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Object Oriented Programming	1. Describe the uses of enumerated types. 2. Identify and describe the parts of an object class and/or structure. 3. Apply object methods to illustrate real world objects. 4. Compare and contrast objects and addresses/references to objects.	4	4	0
Numerical Methods	1. Apply Linear Interpolation to a real world problem and design a solution. 2. Apply Linear Regression to a real world problem and design a solution. 3. Apply Pseudo Random Numbers to a real world problem and design a solution. 4. Apply the Bisection Method to a real world problem and design a solution. 5. Apply Gaussian Elimination to a real world problem and design a solution. 6. Apply the Gauss-Seidel method to a real world problem and design a solution. 7. Apply Numerical Integration to a real world problem and design a solution.	8	8	0

Total Contact Hours

Lecture Hours	Lab Hours	Clinical Hours
30	30	0