Glendale College

Course Outline of Record Report

Course ID 005234

Revision - May 2025

CS/IS280 : Concepts Of Programming Languages

General Information

Author:	• Tony Biehl
Attachments:	DE Addendum_CS:IS_280(CE)_COR_8:1:20_CoDE_2:27:24.pdf
Course Code (CB01) :	CS/IS280
Course Title (CB02) :	Concepts Of Programming Languages
Department:	CSIS
Proposal Start:	Spring 2026
TOP Code (CB03) :	(0706.00) Computer Science (transfer)
CIP Code:	(11.0701) Computer Science.
SAM Code (CB09) :	E - Non-Occupational
Distance Education Approved:	No
Will this course be taught asynchronously?:	No
Course Control Number (CB00) :	CCC000094643
Curriculum Committee Approval Date:	05/28/2025
Board of Trustees Approval Date:	07/08/2025
Last Cyclical Review Date:	09/01/2020
Course Description and Course Note:	CS/IS 280 discusses issues in the design, implementation and use of high-level programming languages, historical background, and how languages reflect different design philosophies and user requirements. Technical issues in the design of major procedural programming languages and other approaches to programming languages, such as functional programming, logic programming, and object oriented programming, are studied.
Justification:	Transferability/C-ID Change
Academic Career:	• Credit
Mode of Delivery:	• In-Person • Remote • Hybrid • Online
Author:	No value
Course Family:	No value

Academic Senate Discipline

Primary Discipline:	• Computer Science
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 with Pass / No-Pass Option

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
ITIS	Information Technology and Information Systems	Approved	08/30/2021	ITIS 130 - Introduction to Programming Concepts and Methodologies

Units and Hours

Summary

Minimum Credit Units (CB07)	3
Maximum Credit Units (CB06)	3
Total Course In-Class (Contact) Hours	54
Total Course Out-of-Class Hours	108
Total Student Learning Hours	162

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

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

Course Student Hours

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

CS/IS135 - Programming In C/C++

Objectives

- Examine problems, apply logic, and provide solutions/algorithms for the problems.
- Recognize programming problems on a function-by-function basis and develop structured/procedural code based on this approach.
- Demonstrate an understanding of object-oriented programming concepts and object-oriented design in creating a program.
- Program in the C++ language including use of objects, pointers, and structures.

Entry Standards

Entry Standards	Description
No value	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	
Methods of Instruction	Lecture
Methods of Instruction	Demonstrations

Out of Class Assignments

Homework exercises

- Demonstrate the practical differences between the major programming languages.
- Programming projects like implementing Abstract Data Types

Methods of Evaluation	Rationale
Exam/Quiz/Test	Final examination.
Exam/Quiz/Test	midterm examinations and quizzes;

Textbook Rationale

No Value

Textbooks

Author	Title	Publisher	Date	ISBN
Sebesta, Robert W,	Concepts of Programming Languages.	Prentice Hall	2019	978-0134997186

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

No Value

Learning Outcomes

Course Objectives

Explain the need for higher level programming languages.

Describe the issues in designing languages.

Create and run simple programs in a variety of computer languages.

Describe mechanisms used in allocating data types and passing data.

Explain how a language is defined and parsed.

SLOs

Explain the evolution of the major programming languages Expected Outcome Performance: 70.0

CSIS Computer Programmer - Certificate	Analyze a programming task/problem; based on that analysis, design and implement an object oriented program using multiple classes in a high level language.
ILOs Core ILOs	Communicate clearly, ethically, and creatively; listen actively and engage respectfully with others; consider situational, cultural, and personal contexts within or across multiple modes of communication.
CSIS Information Technology Certificate	Demonstrate installing, configuring and maintaining computer and mobile devices, including diagnosing, resolving and documenting common hardware and software.
CSIS Information Technology - A.S. Degree Major	Demonstrate installing, configuring, and maintaining computer and mobile devices, including diagnosing, resolving, and documenting common hardware and software.

CSIS Computer Science - A.S. Degree Major	Prepare a software project to implement a single scientific, mathematical, business, or technical function.
CSIS Computer Science - Certificate	Prepare a software project to implement a single scientific, mathematical, business, or technical function.
CSIS Computer Software Technician	demonstrate the ability to independently create, save, modify and print a document using a word processing program and appropriate assistive technology
	write a computer program using either C/C++, Java, or Visual Basic
CSIS Web Development - A.S. Degree Major	use industry standard tools and techniques to produce, publish and maintain Web sites and Web content.
CSIS Web Development - Certificate	use industry standard tools and techniques to produce, publish and maintain Web sites and Web content.

Explain and implement subprograms Expected Outcome Performance: 70.0

CSIS Computer Programmer - Certificate	Analyze a programming task/problem; based on that analysis, design and implement an object oriented program using multiple classes in a high level language.
ILOs Core ILOs	Communicate clearly, ethically, and creatively; listen actively and engage respectfully with others; consider situational, cultural, and personal contexts within or across multiple modes of communication.
CSIS Information Technology Certificate	Demonstrate installing, configuring and maintaining computer and mobile devices, including diagnosing, resolving and documenting common hardware and software.
CSIS Information Technology - A.S. Degree Major	Demonstrate installing, configuring, and maintaining computer and mobile devices, including diagnosing, resolving, and documenting common hardware and software.
CSIS Computer Science - A.S. Degree Major	Prepare a software project to implement a single scientific, mathematical, business, or technical function.
CSIS Computer Science - Certificate	Prepare a software project to implement a single scientific, mathematical, business, or technical function.
CSIS Computer Software Technician	demonstrate the ability to independently create, save, modify and print a document using a word processing program and appropriate assistive technology
	write a computer program using either C/C++, Java, or Visual Basic
CSIS Web Development - A.S. Degree Major	use industry standard tools and techniques to produce, publish and maintain Web sites and Web content.
CSIS Web Development - Certificate	use industry standard tools and techniques to produce, publish and maintain Web sites and Web content.

Demonstrate how an interpreter would work Expected Outcome Performance: 70.0

CSIS Computer Programmer - Certificate	Analyze a programming task/problem; based on that analysis, design and implement an object oriented program using multiple classes in a high level language.
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.

CSIS Information Technology Certificate	Demonstrate installing, configuring and maintaining computer and mobile devices, including diagnosing, resolving and documenting common hardware and software.
CSIS Information Technology - A.S. Degree Major	Demonstrate installing, configuring, and maintaining computer and mobile devices, including diagnosing, resolving, and documenting common hardware and software.
CSIS Computer Science - Certificate	Prepare a software project to implement a single scientific, mathematical, business, or technical function.
CSIS Computer Science - A.S. Degree Major	Prepare a software project to implement a single scientific, mathematical, business, or technical function.
CSIS Computer Software Technician	demonstrate the ability to independently create, save, modify and print a document using a word processing program and appropriate assistive technology
	write a computer program using either C/C++, Java, or Visual Basic
CSIS Web Development - A.S. Degree Major	use industry standard tools and techniques to produce, publish and maintain Web sites and Web content.
CSIS Web Development - Certificate	use industry standard tools and techniques to produce, publish and maintain Web sites and Web content.

Additional SLO Information

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

Yes

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

No

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

It is to make a more consistent prerequisite across several upper-division CS courses. That would aid students in taking CS courses

SLO Evidence

No Value

Course Content

Lecture Content

Introduction (3 Hours)

- Objective of course
- Needed software tools

Evolution of the Major Programming Languages (4 Hours)

- First programming languages
- Issues with subsequent languages
- Current languages overview

Describing Syntax and Semantics (4 Hours)
- How the language is coded
- What does the code actually do

Lexical and Syntax Analysis (4 Hours)
- How to break apart a statement
- How to ensure syntax is correct

Names, Binding, Type Checking, and Scopes (3 Hours)
- How to manage identifiers and bind to appropriate resource
- How to ensure correct types
- How to resolve where and how identifier are accessed

Data Types (3 Hours)
- Specify data types language allows
- Specify operations with the data types

Expressions and Assignment Statements (3 Hours)
- Ensure correct operations for particular data types
- Ensure correct assignment to an identifier and any converting of values allowed

Statement-Level Control Structure (3 Hours)
- The if statements
- Selection statements
- Looping control

Subprogram (4 Hours)
- How subprograms are defined
- How subprograms are used

Implementing Subprograms (4 Hours)
- Syntax of subprograms
- How data is passed and returned from subprograms
- Are nested subprograms allowed

Abstract Data Types (ADT) (4 Hours)
- How user can defined own data types (ADT)
- How language used the ADT

Support for Object-Oriented Programming (3 Hours)
- Syntax of describing classes
- How objects are created

Concurrency (3 Hours)
- Advantages of more than one routine running at once
- Disadvantages and synchronization problems

Exception Handling (3 Hours)
- Kinds of errors that are severe
- Implementing mechanisms for error handling

Functional Programming Language (3 Hours)
- How to deal with a functional programming language
- Example languages

Logic Programming Languages (3 Hours)
- Uses of logic programming
- Example languages

Total Hours = 54

Additional Information

Repeatability

Not Repeatable

Justification (if repeatable was chosen above)

No Value

Is it possible this course will have a material fee?

No

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

No

What term(s) will this course be offered?

No Value

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

- No

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

No Value