

Glendale College Course Outline of Record Report

Course ID 005209
Revision - March 2025

CS/IS151 : Python Programming

General Information

Author:	<ul style="list-style-type: none"> • Edwin Sookiassian • Biehl, Tony
Course Code (CB01) :	CS/IS151
Course Title (CB02) :	Python Programming
Department:	CSIS
Proposal Start:	Spring 2026
SAM Code (CB09) :	C - Clearly Occupational
Distance Education Approved:	No
Will this course be taught asynchronously?:	Yes
Course Control Number (CB00) :	CCC000502166
Curriculum Committee Approval Date:	03/26/2025
Board of Trustees Approval Date:	06/17/2025
Last Cyclical Review Date:	03/26/2025
Course Description and Course Note:	CS/IS 151 is a course in programming computers in the Python language for students interested in programming, graphics, and Graphical User Interface (GUI) programming in business, machine learning, and game applications. Students are introduced to the basics of the Python language and review computer science concepts such as data types, decision structures, loops, functions, and object-oriented programming.
Justification:	Content Change
Academic Career:	<ul style="list-style-type: none"> • Credit
Mode of Delivery:	No value
Author:	<ul style="list-style-type: none"> • Edwin Sookiassian • Biehl, Tony
Course Family:	No value

Academic Senate Discipline

Primary Discipline:	<ul style="list-style-type: none"> • 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)

Local GE Requirement

Transferability

Transferable to both UC and CSU

Transferability Status

Approved

GCC General Education Requirements

Area 2: Mathematical Concepts and Quantitative Reasoning

Area

Mathematical Concepts and Quantitative Reasoning

Status

Approved

Approval Date

09/02/2025

Comparable Course

No Comparable Course defined.

Units and Hours

Summary

Minimum Credit Units (CB07)	3
Maximum Credit Units (CB06)	3
Total Course In-Class (Contact) Hours	90
Total Course Out-of-Class Hours	72
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.

Funding Agency Category (CB23)

Not Applicable.

Cooperative Work Experience Education Status (CB10)

Variable Credit Course

Weekly Student Hours

	In Class	Out of Class
Lecture Hours	2	4
Laboratory Hours	3	0
Studio Hours	0	0

Course Student Hours

Course Duration (Weeks)	18
Hours per unit divisor	54
Course In-Class (Contact) Hours	
Lecture	36
Laboratory	54
Studio	0
Total	90
Course Out-of-Class Hours	
Lecture	72
Laboratory	0
Studio	0
Total	72

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

No Value

Entry Standards

Entry Standards	Description
No value	No value

Course Limitations	
Cross Listed or Equivalent Course	Description
No value	No value

Specifications	
Methods of Instruction	
Methods of Instruction	Lecture
Methods of Instruction	Multimedia
Methods of Instruction	Demonstrations

Out of Class Assignments
<ul style="list-style-type: none"> • Programming assignments (e.g. designing a playing card game) • Computer assignments (e.g. hands-on exploration of GUI programming fundamentals)

Methods of Evaluation	Description of Activity/Interaction
Exam/Quiz/Test	Final Examination
Exam/Quiz/Test	Quizzes
Exam/Quiz/Test	Midterm Examination

Textbook Rationale
No Value

Textbooks				
Author	Title	Publisher	Date	ISBN
Gaddis, Tony	Starting Out with Python	Pearson	2023	9780137871209

Other Instructional Materials (i.e. OER, handouts)
No Value

Learning Outcomes

Course Objectives

Use basic programming concepts.

Code complete programs from program descriptions and provide complete documentation.

Acquire a vocabulary of Python commands.

Develop and code intermediate-level object-oriented programs using Python.

Describe and implement basic graphics in Python programs.

SLOs

Explain, recognize and describe Python programming commands and code.

Expected Outcome Performance: 70.0

Create intermediate level object oriented Python code.

Expected Outcome Performance: 70.0

Describe and implement basic graphical user interface programs in Python.

Expected Outcome Performance: 70.0

Course Content

Lecture Content

Introduction to the Python Language (4 Hours)

- Overview of the software development life-cycle (design, development, documentation, testing, and maintenance)
- Compiled vs. interpreted languages (the Python interpreter)
- Brief history and survey of current object-oriented languages
- Program design tools and programming environments

Writing Programs: Basic Python Syntax (2 Hours)

- Software development
- Elements of programs: names, expressions, output statements
- Computing with numbers: numeric concepts, data types, variables, assignment statements, simple Input/Output (I/O)

Computing with Strings (2 Hours)

- Python syntax
- Coding conventions
- String operations: indexing, slicing, string conversion and formatting tools

Basic Control Structures (8 Hours)

- Algorithms and selection structures
- If statements and multi-way branching (if/elif) and block delimiters
- Boolean operators
- Loops and repetition statements

Intermediate Control Structures for Loops and Tuples (2 Hours)

- Iteration with for loops and sequence traversal
- Tuple creation, immutability, and practical usage examples

Python Collection Types: Lists and Dictionaries (3 Hours)

- Arrays and multi-dimensional arrays
- Sorting and searching algorithms

Functions: Definitions and Calls; Scopes and Arguments (5 Hours)

- Passing parameters by value and by reference
- Documentation of functions

Files, Streams, and Exceptions (2 Hours)

- Sequential access
- Principles of testing and designing test data

Software Objects/Object-Oriented Programming (8 Hours)

- Using software objects
- Object-oriented concepts Designing and implementing object-oriented programs
- Documentation of classes and their members
- Basic graphics and graphical user interfaces (GUI)
- Basic graphics and GUI programming concepts
- Basic program development using the Tkinter graphics module

Total Hours: 36**Laboratory/Studio Content****Labs (54 Hours)**

- if
- if-else
- Nested if and if-else
- while
- for
- Methods/functions
- Single Dimensional Arrays
- Multi-Dimensional Arrays
- Objects and classes
- Git

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 Value

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

No Value

What term(s) will this course be offered?

Spring

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

Resources

Did you contact your departmental library liaison?

No

If yes, who is your departmental library liaison?

No Value

Did you contact the DEIA liaison?

No

Were there any DEIA changes made to this outline?

No Value

If yes, in what areas were these changes made:

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