CSS - 190A - Python Fundamentals (Fall 2025)

01. UG Omnibus or Individual Variable Topic Course (New or Modification)

Due Dates and Resources

NOTE: Omnibus courses may be offered a maximum of three times.

If you have questions or need assistance in filling out this proposal form, you may contact the Office of Curriculum.

Deadlines for curriculum can be found:

- Curriculum SharePoint
- Curriculum Website
- Procedural Calendar
- On your Curriculog dashboard under 'My Upcoming Events'

In order to meet the deadline, this proposal must be on the *Simultaneous Department/Associate Dean Review* step on or before the listed due date.

Resources for curriculum can be found:

- Originator How-To Guide
- Curriculum SharePoint

This form SHOULD be used for the following:

- Creating a new undergraduate omnibus course
- Creating a new undergraduate variable topics course
- Modifying an existing undergraduate omnibus course
- Modifying an existing undergraduate variable topics course

This form SHOULD NOT be used for the following:

- Creation of a new permanent course
- Modifying an existing permanent course
- Creation or modification of a graduate omnibus or variable topics course

Directions for Form

General Instructions and Information

You may collapse individual sections of this form by clicking the arrow or "V" icon to the right of the section title.

All fields that are marked with an asterisk (*) are required.

Each section may have additional directions attached. Please follow all instructions.

Note: Proposals that are incomplete or filled out incorrectly will be returned to the originator.

INSTRUCTIONS FOR CREATING A NEW COURSE

Fill out all of the below fields. New courses can skip Part II.
Launch the proposal.
Approve the proposal.

Use the checkmark icon on the right of the screen to approve the proposal.

INSTRUCTIONS FOR MODIFYING AN EXISTING COURSE

Import the course you wish to modify.
Fill out Part I of the form.
LAUNCH the proposal.
Fill out Part II to indicate all the modifications you make to the course.
Modify any course fields in Part III as needed.

Modification Note: DO NOT MAKE CHANGES TO YOUR COURSE INFORMATION (PART III) UNTIL AFTER YOU LAUNCH THE PROPOSAL in order to track changes.

Failure to use the track changes feature may cause a delay or denial of your proposal.

If you modify additional fields not already indicated in Part II, please make sure you add them to the modification list.

Approve the proposal.

Use the checkmark icon on the right of the screen to approve

approve

the
proposal.

Part I: Fill out BEFORE launching the proposal

College/School:* College of Aerospace, Computing, Engineering, and Design

Department:* Department of Computer Sciences

Name of Proposal Originator:* Yiming Zhao

Email of Proposal Originator:* yizhao@msudenver.edu

Is this a new course or modification of an existing course?*
- (•) New Omnibus Course
- ( ) Modification of Omnibus Course
- ( ) New Individual Variable Topic Course
- ( ) Modification of Individual Variable Topic Course

If an omnibus course, will Senior Experience designation be requested?*
- (•) No
- ( ) Yes
- ( ) N/A It's an Individual Variable Topic Course

If you select "Yes" to the Omnibus Senior Experience designation question, ensure you complete the Senior Experience Designation section of this form and follow all the rules regarding this designation.

Please indicate the semester you wish this change to be effective. You may view curriculum due date information at the top of this page. If you select a semester that the due dates have passed for, it will be implemented the following semester.

You must also put the semester and year you want the course to run in the course title field - example: (Fall 2019).

Indicate Effective Semester:* ( ) Spring ( ) Summer (•) Fall

Justification and resource implication for proposed curriculum action*

Proposed Change:

We propose a new course that focuses on Python programming language.

Rationale

The need for this proposed Python programming course arises from recurring inquiries both within and outside our department regarding the availability of such a course. Over time, CS department has received multiple queries from students and other departments about whether we offer Python programming instruction, highlighting a clear demand for this subject. Additionally, the diverse backgrounds of incoming Computer Science freshmen justify the creation of this course. While some students enter with prior programming experience and can successfully begin with CS1050 (Java), others find it challenging to do so. Moreover, some CS majors are capable of independently learning Python after completing CS1050/2050, while others require a structured course to effectively develop proficiency in Python programming.

This course is created to address these diverse needs. Incoming CS freshmen and current CS majors could benefit from such a structured Python instruction. This course is not required in CS major and doesn't have any prerequisites. Our CS general advisors will guide interested students toward enrolling in the course, ensuring that those who need it receive proper support without making it a mandatory part of the CS major program.

Furthermore, students in the Data Science and Machine Learning (DSML) major may exhibit similar learning needs. By offering this Python course, we aim to provide an accessible and well-supported learning pathway for students across multiple disciplines who require Python programming skills for their academic and professional development.

In addition to serving CS and DSML students, this course will also be open to students from other departments who are interested in learning Python. Python is extensively used across various disciplines, including engineering, mathematics, business, biology, and social sciences, making it a valuable skill for students beyond computer science. By offering an accessible and well-structured introduction to Python, this course can support interdisciplinary learning and equip students with basic programming skills applicable to their respective fields. This Python programming course benefits not only students but also the department, college, and university as a whole.

Resource implication

Academic advisors need to update advising materials to reflect this new course. This ensures that students are well-informed about the availability and purpose of the course. Academic advisors can recommend students without prior CS experience and students from other majors who wish to learn Python programming to take this course.

According to the Undergraduate Curriculum Manual, it is the responsibility of both the originator as well as each level of review to consider potential overlap and curriculum conflict. Any potential overlap or conflict with existing curriculum should be reviewed, and the impacted department(s) should be requested to provide a letter of notification or support, depending on the circumstances. Attach documentation that supports affected Departments were notified and/or provided support of the proposed changes in the Proposal Toolbox by clicking on the paperclip icon on the right side of the form.

Please Confirm That:* ☑ I, the originator of this proposal, have completed the necessary due diligence to review this proposal for any potential overlap and/or conflict with existing curriculum. Any departments identified as having potential overlap and/or conflicts have been contacted and a letter of notification and/or a letter of support has been obtained.

Part II: Course Modification Information

If modifying an existing course, please specify which sections have been modified (check all that apply):

- ☐ Part IIIa (prefix, course number, course title, transcript/banner course title, course type, CIP code)
- ☐ Part IIIb (credits, distribution of credits, schedule type, grade mode, contact hours, repeats, equivalencies, crosslistings, registration restrictions)
- ☐ Part IIIc (prerequisites, corequisites, or prerequisite(s)/corequisite(s), banner enforced prerequisites, corequisites, or prerequisite(s)/corequisite(s), catalog course description)
- ☐ Part IIId (required reading, SBLOs, outline, evaluation of student performance)

Reminder if you are making modifications to this course:

DO NOT MAKE CHANGES TO YOUR COURSE INFORMATION (PART IIIa-IIId) UNTIL AFTER YOU LAUNCH THE PROPOSAL in order to track changes. Failure to use the track changes feature may cause a delay or denial of your proposal.

Part IIIa: Course Information

Prefix:* CSS

Course Number:* 190A

Course Title (include semester and year for course to run):* Python Fundamentals (Fall 2025)

Transcript/Banner Course Title:* Python Fundamentals

Course Type:* Computer Science Studies

Prefix, Number, and Title of Umbrella Course:

CIP Code: 11.0701

Part IIIb: Course Information, continued

To receive Title IV financial aid funds, all institutions of higher education must comply with the federal definition of a credit hour. The Higher Learning Commission requires institutions to maintain policies and procedures for verifying compliance with this definition.

Federal Credit Hour Definition: *A credit hour is an amount of work represented in intended learning outcomes and verified by evidence of student achievement that is an institutionally-established equivalency that reasonably approximates not less than:*
(1) one hour of classroom or direct faculty instruction and a minimum of two hours of out-of- class student work each week for approximately fifteen weeks for one semester or trimester hour of credit, or ten to twelve weeks for one quarter hour of credit, or the equivalent amount of work over a different amount of time; or (2) at least an equivalent amount of work as required in paragraph (1) of this definition for other activities as established by an institution, including laboratory work, internships, practica, studio work, and other academic work leading toward to the award of credit hours. 34CFR 600.2 (11/1/2010)

Credits:* 3

Distribution of Credits:* 3+0

Schedule Type(s):* Lecture

Grade Mode(s):* Letter

Face-to-Face or Equivalent Hours per course

Consult Appendix B and C of the Curriculum Manual to determine the hours for the course

Lecture: 37.5

Lab:

Internship:

Practicum:

Other Hours:

Additional Student Work Hours: 90

Please answer yes or no to the below questions. If you answer yes to any of the questions, please fill out the related field on the right.

A specified repeatable course is a course that allows a student to repeat the course either in its entirety or for a certain identified total number of credit hours. If you decide to make your course repeatable, please specify either how many times a student can repeat the course for credit, or for the total number of credits they can receive.

Is this course a specified repeatable course?* (•) No ( ) Yes

If yes, indicate specified repeatable number of credits and/or repeats allowed:

A crosslisting is when a course is made available under additional prefixes for students in other programs.

An equivalency is when two courses are coded in Banner to be equal to each other.

Generally equivalencies are used when an old, archived course is needed to be equal to a new course. Crosslistings are used for active courses. Supporting documentation should be included to demonstrate approval for crosslistings.

Are there course equivalencies?* (•) No ( ) Yes

If yes, list all equivalent courses in alphabetical order:

Are there course crosslistings?* (•) No ( ) Yes

If yes, list all crosslistings in alphabetical order:

Registration Restrictions

Program:

Major:

Level:

Class:

Student Attribute:

Part IIIc: Course Information, continued

The following fields will allow you to attach prerequisites, corequisites, or prerequisites or corequisites to your course.

Omnibus courses will, by default, have ALL prerequisites, corequisites, and prerequisites or corequisites Banner enforced. **Therefore, you must ensure that all of your prerequisites, corequisites, or prerequisites or corequisites are listed as Banner enforced.**

All the following fields regarding prerequisites, corequisites, and prerequisites or corequisites are REQUIRED. If you do not wish to have anything in the field, please type None.

Please also indicate the minimum passing grade.

Prerequisite(s):* N/A

Banner Enforced Prerequisite(s):* N/A

Minimum Passing Grade for Banner Enforced Prerequisite(s):* N/A

Corequisite(s):*	N/A
Banner Enforced Corequisite(s):*	N/A
Prerequisite(s) or Corequisite(s):*	N/A
Banner Enforced Prerequisite(s) or Corequisite(s):*	N/A
Minimum Passing Grade for Banner Enforced Prerequisite(s) or Corequisite(s):*	N/A

Course Description:*

In this course, students study how to identify fundamental concepts of the Python programming language and how to apply basic programming techniques to solve introductory computing problems. Emphasis is placed on building fundamental programming abilities, applying core concepts such as variables, loops, conditionals, and functions, and developing a structured approach to solving introductory computing problems. Students practice breaking down problems into manageable steps and constructing clear, efficient solutions using Python. Through hands-on practice, students explore real-world applications of Python and observehow programming can be utilized to solve practical, introductory problems across various domains. By the end of the course, students have the foundational knowledge necessary to construct basic programs and adapt their skills to more advanced computational challenges.

Part IIId: Course Information, continued

The following section is the course content. You must adhere to the following format for each section:

Required reading: Please list materials in preferred citation style (eg. MLA, APA, etc.).

List each material in this format. If there are multiple materials, format them in a bullet or list style

Specific Measurable Student Behavioral Learning Objectives: Please list the SBLOs in your preferred numbering or bulleting style. Start section with: Upon completion of this course, the student should be able to:.
Detailed Outline of Course Content or Outline of Field Experience/Internship: Please list the course outline in your preferred numbering or bulleting style. It is recommended that you use a numbering format for this field.
Evaluation of Student Performance: Please list the evaluation of student performance in your preferred numbering or bulleting style.

You must use the numbering list feature within the toolbar above each field. Right click on a number in the list and select "Numbered List Properties" to change the numbering style. Please maintain consistency in the selected numbering or bulleting styles.

Reminder if you are making modifications to this course:

DO NOT MAKE CHANGES TO YOUR COURSE INFORMATION (PART IIIa-IIId) UNTIL AFTER YOU LAUNCH THE PROPOSAL in order to track changes. Failure to use the track changes feature may cause a delay or denial of your proposal.

Required reading and other materials will be equivalent to:*

Some recommended textbooks are listed below. They are equivalent to each other. The instructor can also choose their preferred textbook.

1. Whitington, John. Python from the Very Beginning: with more than 100 exercises and answers. Coherent Press, 2020.
2. Zingaro, Daniel. Learn to code by solving problems: A Python programming primer. No Starch Press, 2021.
3. Ramalho, Luciano. Fluent Python. O'Reilly Media. 2023.

Specific, Measurable Student Behavioral Learning Objectives:*

Upon completion of this course, the student should be able to:

1. Demonstrate an understanding of programming in Python.
2. Write and debug Python functions using parameters and return values.
3. Construct and analyze Python programs that use expressions, assignment, decision structures, and loops to solve problems.
4. Manipulate and implement basic data structures such as lists and tuples to store and process information.
5. Apply best practices in Python programming, such as modular design, readability, and maintainability, to write clear and structured code.
6. Incorporate built-in and external Python modules (e.g., math, random, time) to enhance program functionality and expand programming capabilities.
7. Utilize Python libraries for scientific computing and introductory data analysis, including generating basic visualizations with Matplotlib, applying NumPy for numerical computations and array operations, using SymPy for symbolic mathematics and algebraic manipulations, and employing Pandas for data analysis and structured data handling.
8. Develop Python programs to solve simple computational problems, such as calculating the sum of values in a list.
9. Observe how Python can be used to implement certain software features, such as developing a simple graphical user interface (GUI).

Detailed Outline of Course Content (Major Topics and Subtopics) or Outline of Field Experience/Internship *

1. Introduction to Computers and Programming
    1. overview of computer systems, programming languages, and development environments
    2. introduction to Python and its role in computing
    3. problem-solving and modular design in programming
2. Python Basics
    1. understanding Python syntax and semantics
    2. variables, identifiers, and data types (integers, floats, strings, Boolean values)
    3. expressions, operators, and operator precedence
3. Decision Structures
    1. logical expressions and operators (not, and, or)
    2. conditional statements: if, if-else, if-elif-else
    3. ternary operator
    4. nested decision structures
4. Looping Structures
    1. while loops and for loops
    2. loop control statements (break, continue)
    3. avoiding infinite loops
5. Functions and Modular Programming
    1. defining and calling functions
    2. function parameters and return values
    3. scope and visibility of variables
6. Working with Lists
    1. list creation and basic operations
    2. iterating through lists using loops
    3. list methods and basic list processing
    4. 2D lists
7. Utilizing Python Modules
    1. introduction to built-in modules (math, random, time)
    2. importing and using module functions
8. Input and Output Operations
    1. interactive input using input()
    2. displaying output with print()
    3. file input/output operations and writing text files

3. file input/output: reading and writing text files

9. Introduction to Problem Solving with Python
    1. developing simple programs to solve easy computing problems, e.g., calculating the sum of all values in a list
    2. breaking problems into smaller, manageable parts
    3. writing structured and maintainable code
10. Introductory Scientific Computing and Data Analysis
    1. utilizing NumPy to perform numerical computations, e.g., creating arrays and applying mathematical operations
    2. manipulating symbolic expressions using SymPy, e.g., simplifying algebraic expressions and solving equations
    3. handling and analyzing structured data with Pandas, e.g., organizing datasets and performing basic statistical operations
    4. creating visual representations of data using Matplotlib, e.g., generating line charts, bar graphs, and scatter plots
11. Observing Python Applications in Software Development
    1. demonstrations of Python in real-world applications, e.g., graphical user interfaces (GUIs) in Python
    2. exploring how small code modifications impact program behavior

Evaluation of Student Performance:*

The evaluation of student performance will be determined by the instructor and may include, but is not limited to, the following components. If possible, assignments and exams should reflect how programming can be applied to real-world scenarios, emphasizing its impact on everyday life. This approach can help motivate students without prior computer science experience to explore computing as a potential career path.

1. Homework and programming assignments
2. Quizzes and examinations
3. Final examination
4. Small individual or group projects involving problem-solving with Python or modifying existing Python programs
5. Oral presentations

Part IV: Senior Experience Designation

In light of the extraordinary circumstances we are currently facing, omnibus courses will temporarily be able to submit for the Senior Experience designation. By selecting the Senior Experience Designation option:

> **You are required to fill in the below fields that describes how this course will fulfill the Senior Experience criteria. Please be detailed.**
> **You are required to have Senior Standing as a prerequisite for this course, and you must enforce this via the registration restrictions. You should select "senior" in the class registration restriction. This is required of all Senior Experience courses.**
> **The course should be a 490 course number.**
> **The Faculty Senate Curriculum Committee will review your omnibus course to ensure that appropriate rigor is present and the Senior Experience criteria and requirements are fulfilled.**
>
> > **As the originator of the proposal, you will be listed on the step with Faculty Senate Curriculum Committee in case changes are requested. This means you will need to APPROVE on this step.**
>
> **Even if you submit your Senior Experience request on an omnibus course, it MUST be submitted by the omnibus deadlines.**

Please describe how this course meets each of the below Senior Experience criteria.

"The Senior Experience must allow students to..."

1. synthesize learning through critical analysis and logical thinking.

Explain

2. apply theoretical constructs to practical applications.

Explain

3. critique philosophical tenets and current practices.

Explain

4. integrate and refine oral and/or written communication skills.

Explain

5. verify their expertise.

Explain

CAEPD and Registrar's Office Use Only

Notes

Form updated May 2024