CSS - 1020 - Python Fundamentals

04. UG Course Modification (No Special Designation)

Due Dates and Resources

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 Modern campus Curriculum dashboard under 'My Upcoming Events'**

In order to meet the deadline, this proposal must be on the *Nonsubstantive, Substantive College/School, OR Substantive University Level Review* step on or before the listed due date. Which step will depend on the changes being made.

Resources for curriculum can be found:

- Originator How-To Guide
- Curriculum SharePoint

This form SHOULD be used for the following:

- Modification of a course **without** a special designation (General Studies, Community Engaged Learning, Ethnic Studies & Social Justice, or Senior Experience).

This form SHOULD NOT be used for the following:

- Modifying a course with a special designation (General Studies, Community Engaged Learning, Ethnic Studies & Social Justice, or Senior Experience).
- Creating a new course with or without a special designation (General Studies, Community Engaged Learning, Ethnic Studies & Social

Justice, or Senior Experience).

Creating or modifying a graduate 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 instructions.

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

INSTRUCTIONS FOR MODIFYING AN EXISTING COURSE

Import the course you wish to modify.
Fill out Part I of the form.

> **Carefully follow the instructions on selecting the level of review your proposal requires. Incorrect review selection will require a resubmission of the proposal.**

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 the proposal.**

Part I: Department and Originator Information (Fill out BEFORE launching the proposal)

If you are changing course ownership, please list both departments and if applicable, both college/schools.

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

Justification and Resource Implication for Curriculum Proposal:*

Proposed Change

This proposal converts the existing omnibus Python programming course into a regular, permanent course.

Justification and Rationale

The omnibus version has been offered multiple times with good enrollment and consistent student interest, demonstrating a clear and ongoing need. Students from Computer Science, Data Science and Machine Learning, and many other majors regularly request a structured introduction to Python, which is widely used in academic, research, and industry settings. Formalizing the course ensures predictable availability and clearer advising guidance.

The permanent course will continue to support students with varied programming backgrounds, including incoming CS students who may benefit from beginning with Python rather than Java. It also provides an accessible pathway for students in other departments who require Python skills for their disciplines. Converting the course to a regular listing strengthens advising, promotes interdisciplinary learning, and aligns with the university's goals of supporting computational literacy.

Resource Implications

Converting the course from an omnibus to a regular offering has minimal resource impact. The department already offers the course using existing resources, space, and instructional materials, and no additional personnel, equipment, or software are needed. The primary change involves updating advising materials and ensuring advisors are aware of the permanent course number and purpose. Since the course has been sustainably taught under the omnibus format, its regularization does not require new resources or letters of support.

Related Curriculum Proposals:* The proposal of the previous omnibus course: CSS - 190A - Python Fundamentals (Fall 2025), https://msudenver.curriculog.com/proposal:12886/form

Impact Report Results:*

Impact Report for CSS 1020

There are no results for this report.

Course Modification Level Review

The modification level review question will determine the workflow of this proposal, so it is essential you select the correct one. Please consult this tool which will help you determine the review level you need. If you select a level of review that does not include the changes you make to the course, this proposal will be denied and you will have to resubmit a new proposal. If you are unsure which level of review you need to select, please contact the Curriculum Staff here.

Note: Changing the level of review after launching the proposal will not change the workflow. If you discover that you have selected an incorrect review level post-launch, you must submit a new proposal. You can contact the Curriculum Office to delete the incorrect proposal.

Course Modification Level Review Selection:* ○ Nonsubstantive ◉ Substantive College/School Level ○ Substantive University Level

Experiential Learning Framework

Review MSU Denver's Experiential Learning Framework website and answer the questions below. Click here to go directly to the framework description.

1. Is this course aligned with MSU Denver's Experiential Learning Framework (this includes community-engaged learning, internships, study abroad, and much more)?* ○ Yes ◉ No

2a. If yes, after reviewing the framework, which tier would you place your course within?
- ☐ Tier 1: Foundational Engagement
- ☐ Tier 2: Intermediate Engagement
- ☐ Tier 3: Advanced Engagement

2b. Explain the rationale for your tier selection:

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

Reminder: This form CANNOT be used to change courses with special designations (Ethnic Studies & Social Justice, Community Engaged Learning, Senior Experience, General Studies). If you submit a course with special designation(s) with this form, it will be denied and you will need to resubmit.

Please indicate on the below list all of the course modifications you are making. **Please do not make changes to sections you do not specify are being modified and make sure to select ALL sections that you modify.**

Specify which sections have been modified (check all that apply):

- [x] 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, catalog note, lab fees, field trips)
- [x] Part IIId (required reading, SBLOs, outline, evaluation of student performance)

Reminder: 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.

If you modify additional sections not already indicated, please make sure you add them to the modification checklist.

Part IIIa: Course Information

Prefix:* CSS

Course Number:* 1020

Course Title:* Python Fundamentals

Transcript/Banner Course Title:* Python Fundamentals

Course Type:* Computer Science Studies

CIP Code: 11.0701

Part IIIb: Course Information, continued

Please check all that apply from the selections below. You may select more than one option if

applicable.

Check All that Apply:* ☐ Required for Major
☐ Required for Minor
☐ Required for Concentration
☐ Required for Certificate
☑ Elective
☐ Specified Elective

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:

Is this course a variable topics umbrella course?* ○ No ○ Yes

If yes, indicate variable topic number of credits and/or repeats available:

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 all active courses. **The crosslisting feature should be used 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. Please specify if you want and of these prerequisites, corequisites, or prerequisites or corequisites Banner enforced.

Banner enforcement means that the requirement will be enforced when the student attempts to register for a course. If you do not Banner enforce the requirement, the system will not check the student's record for the requirement to be met.

Please also indicate the minimum passing grade.

Prerequisite(s): N/A

Banner Enforced Prerequisite(s): N/A

Minimum Passing N/A

Grade for Banner Enforced Prerequisite(s):

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

Catalog 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 observe how programming can be utilized to solve practical, introductory problems across various domains. By the end of the course, students will have the foundational knowledge necessary to construct basic programs and adapt their skills to more advanced computational challenges.

The note field DOES show up in the course listing in the university catalog. A note should be made in specific instances where additional information about a course needs to be conveyed to students. The most common reasons for adding a note are:

The course is crosslisted Example: ***(Note: Credit will be granted for only one prefix.)***
Variable credit courses Example: ***(Note: Variable Credit)***
A course is repeatable Example: ***(Note: This course may be repeated up to 3 times under different topics)*** **OR** ***(Note: This course is repeatable for a maximum of six semester hours)***
If a student cannot take two courses and earn credit for both Example: ***(Note: Students cannot earn credit for XXX1234 and XXX2345)***

Note: N/A

Lab Fees: **Field Trips:**

Part IIId: Course Information, continued

The following section is the course content.

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

List each material in this format. If there are multiple material please 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: 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.

If you modify additional sections not already indicated, please make sure you add them to the modification checklist.

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
    3. observing similarities and differences between Python and other languages commonly used in computer science, such as Java or C++.

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

CAEPD and Registrar's Office Use Only

Notes

Form updated June 2025