# The Create Task

## Create Task

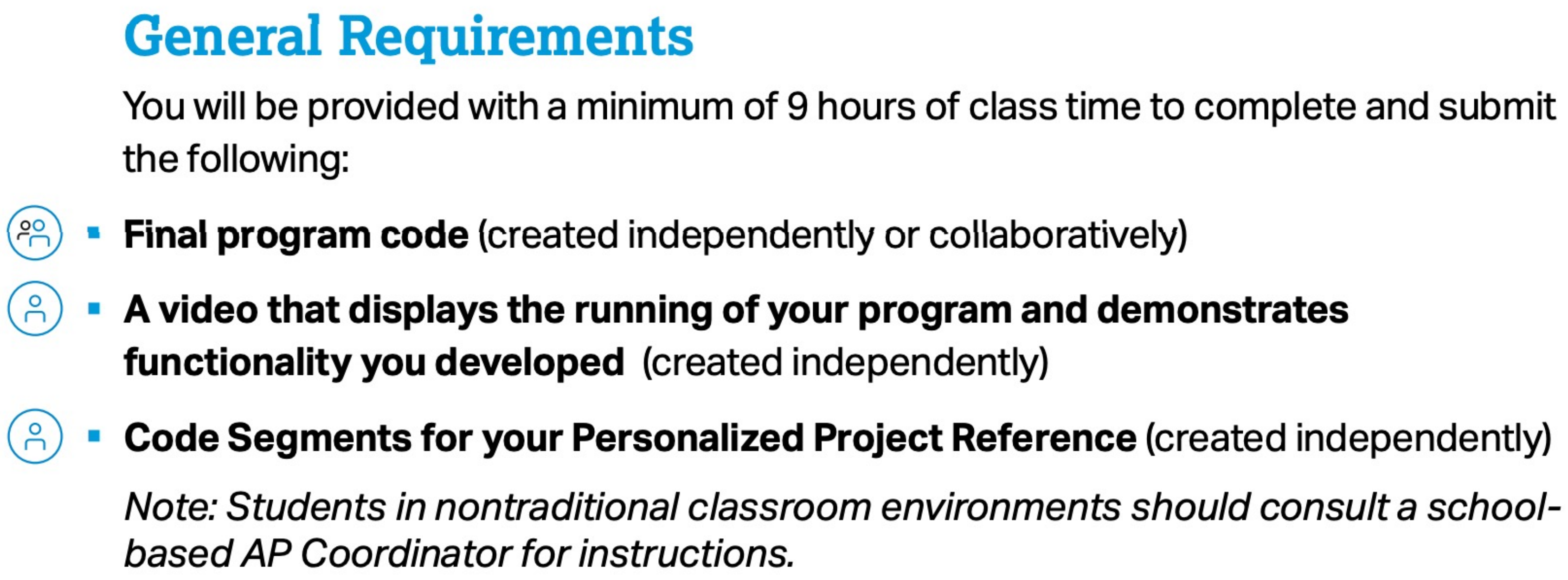
- Written in Python on replit or use Processing
- Game, application, data analysis, or mathematical simulation.

Programming is a collaborative and creative process that brings ideas to life through the development of software. In the Create performance task, you will design and implement a program that might solve a problem, enable innovation, explore personal interests, or express creativity. Your submission must include the elements listed in the Submission Requirements section below.

You are allowed to collaborate with your partner(s) on the development of the program only. The video and Personalized Project Reference that you submit for this performance task must be completed individually, without any collaboration with your partner(s) or anyone else. You can develop the code segments used in your Personalized Project Reference with your partner(s) or on your own as you work on the performance task during class.

Note: Because of all of the citation requirements, I recommend that you work on this task individually. If the program you are writing requires help from others, consider something simpler!

## Requirements



### General Requirements

You will be provided with a minimum of 9 hours of class time to complete and submit the following:

- **Final program code** (created independently or collaboratively)
- **A video that displays the running of your program and demonstrates functionality you developed** (created independently)
- **Code Segments for your Personalized Project Reference** (created independently)

*Note: Students in nontraditional classroom environments should consult a school-based AP Coordinator for instructions.*

## COMPONENT A: PROGRAM CODE (CREATED INDEPENDENTLY OR COLLABORATIVELY)

Submit one PDF file that contains all of your program code (including comments). Include comments or acknowledgments for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

**IMPORTANT:**

If the programming environment allows you to include comments, this is the preferred way to acknowledge and give credit to another author. However, if the programming environment does not allow you to include comments, you can add them in a document editor when you capture your program code for submission.

In your program, you must include student-developed program code that contains the following:

☐ Instructions for input from one of the following:

- the user (including user actions that trigger events)
- a device
- an online data stream
- a file

☐ Use of at least one **list** (or other **collection type**) to represent a collection of data that is stored and used to manage program complexity and help fulfill the program's purpose

**IMPORTANT:**

The data abstraction must make the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).

- ☐ At least one procedure that contributes to the program's intended purpose, where you have defined:
  - the procedure's name
  - the return type (if necessary)
  - one or more parameters

  **IMPORTANT:**

  Implementation of built-in or existing procedures or language structures, such as event handlers or main methods, are not considered student-developed.

- ☐ An algorithm that includes sequencing, selection, and iteration that is in the body of the selected procedure
- ☐ Calls to your student-developed procedure
- ☐ Instructions for output (tactile, audible, visual, or textual) based on input and program functionality

**COMPONENT B: VIDEO** **(CREATED INDEPENDENTLY)** Submit one video file that demonstrates the running of your program as described below. Collaboration is not allowed during the development of your video.

Your video must demonstrate your program running, including:

- ☐ Input to your program
- ☐ At least one aspect of the functionality of your program
- ☐ Output produced by your program

Your video may NOT contain:

- ☐ Any distinguishing information about yourself
- ☐ Voice narration (though text captions are encouraged)

Your video must be:

- ☐ Either .webm, .mp4, .wmv, .avi, or .mov format
- ☐ No more than 1 minute in length
- ☐ No more than 30MB in file size

**COMPONENT C: PERSONALIZED PROJECT REFERENCE** **(CREATED INDEPENDENTLY)** To assist in responding to the written response prompts on exam day, submit required portions of your code by capturing and pasting program code segments you developed during the administration of this task. Screen captures should not be blurry, and text should be at least 10-point font size. Your code segments should not include any comments. These code segments will be made available to you on exam day only if this component is submitted as final in the AP Digital Portfolio by the deadline.

**Procedure:** Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

**IMPORTANT:**

Built-in or existing procedures and language structures, such as event handlers and main methods, are not considered student-developed.

**i.** The first program code segment must be a student-developed procedure that:

- ☐ Defines the procedure's name and return type (if necessary)
- ☐ Contains and uses one or more parameters that have an effect on the functionality of the procedure
- ☐ Implements an algorithm that includes sequencing, selection, and iteration

## Procedure

ii. The second program code segment must show where your student-developed procedure is being called in your program.

**List:** Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.

i. The first program code segment must show how data have been stored in the list.

ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

## Academic Integrity and Plagiarism Policy

This policy addresses plagiarism and academic integrity in completing the Create Performance Task.

## Plagiarism

The use of program code, media (e.g., video, images, sound), data, information, or evidence created by someone else or with generative AI tools in the creation of a program and/or a program code segment(s), without appropriate acknowledgment (i.e., through citation, through attribution, and/or by reference), is considered **plagiarism**. A student who commits plagiarism will receive a score of 0 on the Create performance task, including their responses to the written response prompts on the end-of-course AP Exam.

To the best of their ability, teachers will ensure that students understand how to ethically incorporate ideas that are not their own and provide credit to the original creator or source, as well as the consequences of plagiarism.

## Acceptable Generative AI Use

Students are permitted to utilize generative AI tools as supplementary resources for understanding coding principles, assisting in code development, and debugging. This responsible use aligns with current guidelines for peer collaboration on developing code.

Students should be aware that generative AI tools can produce incomplete code, code that creates or introduces biases, code with errors, inefficiencies in how the code executes, or code complexities that make it difficult to understand and therefore explain the code. It is the student's responsibility to review and understand any code co-written with AI tools, ensuring its functionality. Additionally, students must be prepared to explain their code in detail, as required on the end-of-course AP Exam.

## Preparing for Final Submission

Students are not permitted to collaborate on the video or creation of the Personalized Project Reference.

The Personalized Project Reference cannot include course content or comments within the code or on any other part of the reference. Including course content or comments in the Personalized Project Reference will result in students receiving a score of 0 on the Create performance task, including their responses to the written response prompts on the end-of-course AP Exam.

## Attestations

During the final submission process in the AP Digital Portfolio, students will be asked to attest that they have followed the Performance Task guidelines and have not plagiarized their submission. Each of the three components of the Create performance task must be submitted as final to be sent for scoring. Additionally, if students do not submit their Personalized Project Reference by the deadline, they will not have this resource available on exam day to complete their written response section.

If there is a game that you have always wanted to write, DON'T do it for the create task!

The create task should be as minimal as possible, satisfying all of the requirements and tailored to answer the free response questions!

As we'll see from the examples this week, it can be a very simple program.

# Steps for Create Task

1) Design Program: Brainstorm and think carefully about your program. What is your list? What is the student-developed procedure with parameters that used iteration(loops) and selection(conditionals)? Plan before writing code can save a lot of time!
2) Code the Program: Troubleshoot, test and refine. Start writing the student developed procedure and test it with hard-coded inputs. For example, if you wrote a function called enrypt, call it to make sure it works: print(encrypt("hello", sub)). Refine and repeat.
3) Make Video
4) Make Personalized Project Reference
5) Submit portfolio on College Board site. Make “Final”.

## Timeframe

1) Design including brainstorming, finding datasets(2 hours)
2) Coding(5 hours)(weekly, we will have a set day to work on this)
3) Video(1 hour)
4) Uploading to Digital Portfolio(1 hour)

Total(minimum class time): 9 hours

This is the minimum amount of time you should spend on this. Likely you will need to spend more time(outside of class) to refine and create the best possible portfolio.

## Program Code

Minimum Requirements:

1) Some type of input and output(input can be from keyboard or from file)
2) At least one list(list of tuples, or list of objects, list of Sprites if you use Processing) that manages complexity. The list should be essential to your program. Your program should be very difficult to write without the use of this list.
3) At least one student-developed procedure that has an algorithm which uses one or more parameters that used iteration(loops) and selection(conditionals).
   A) You function must have at least TWO parameters.
   B) In your function, you must have at least two conditionals(if-elif or if-else).

## Create Task Full Example

This sample Create Task program is very simple. It reads data from a text file containing students'names and their GPAs. Depending on the user input, it either prints out the list of names of students whose gpa is below 2.0 or students whose gpa is above 2.0.

This simple program is enough to get full points on the Create Task! The entire program only takes about 20 lines of code!

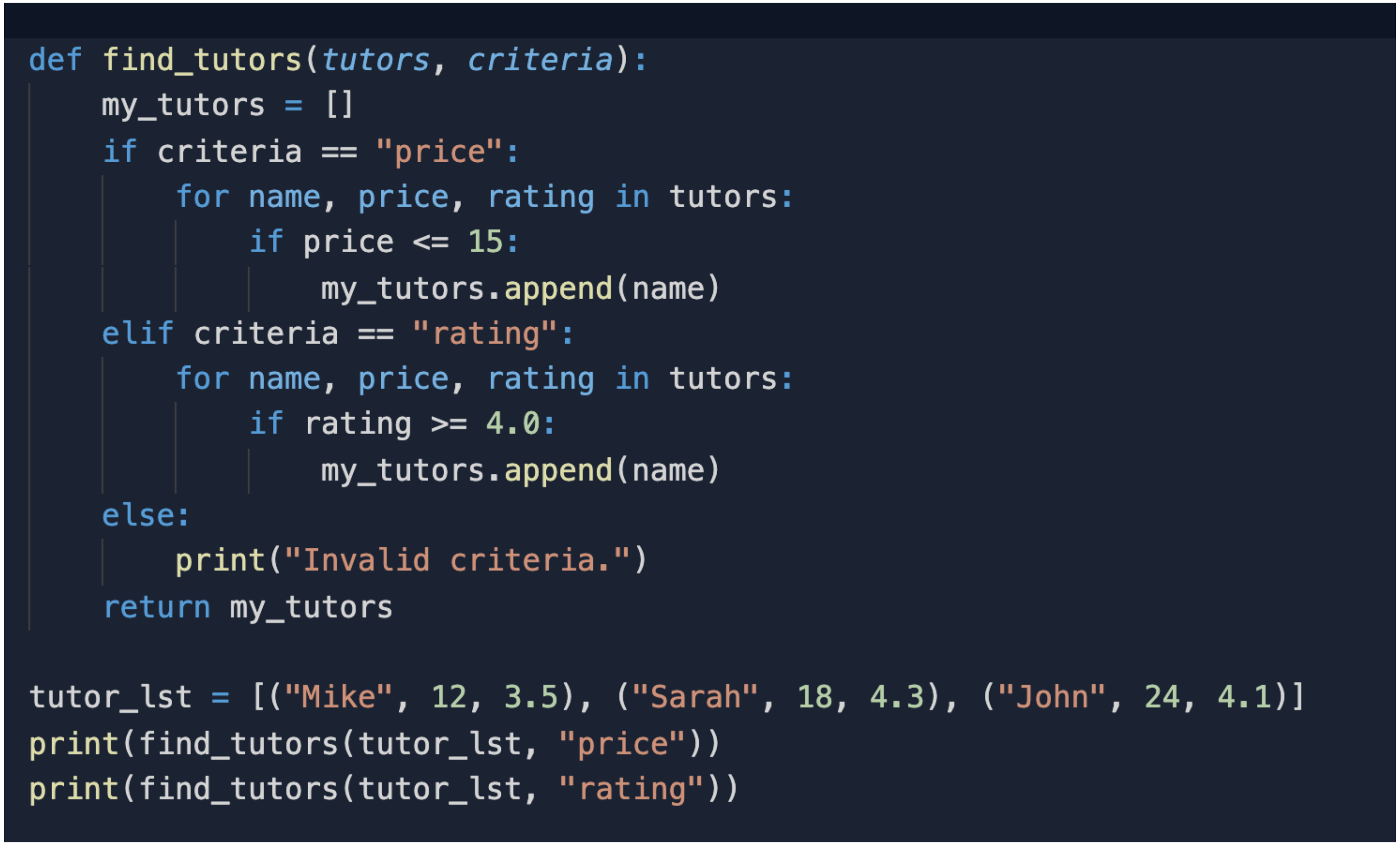
We'll do replit Teams labs that walk through this entire program.

## More Examples

The most essential component of the Create Task is the student-developed function. It is also the trickiest part about the Create Task. If you are thoughtful and plan carefully for this part BEFORE you start coding, it can save a lot of time later on.

The next few slides go over some more examples that can help you come up with your own function.

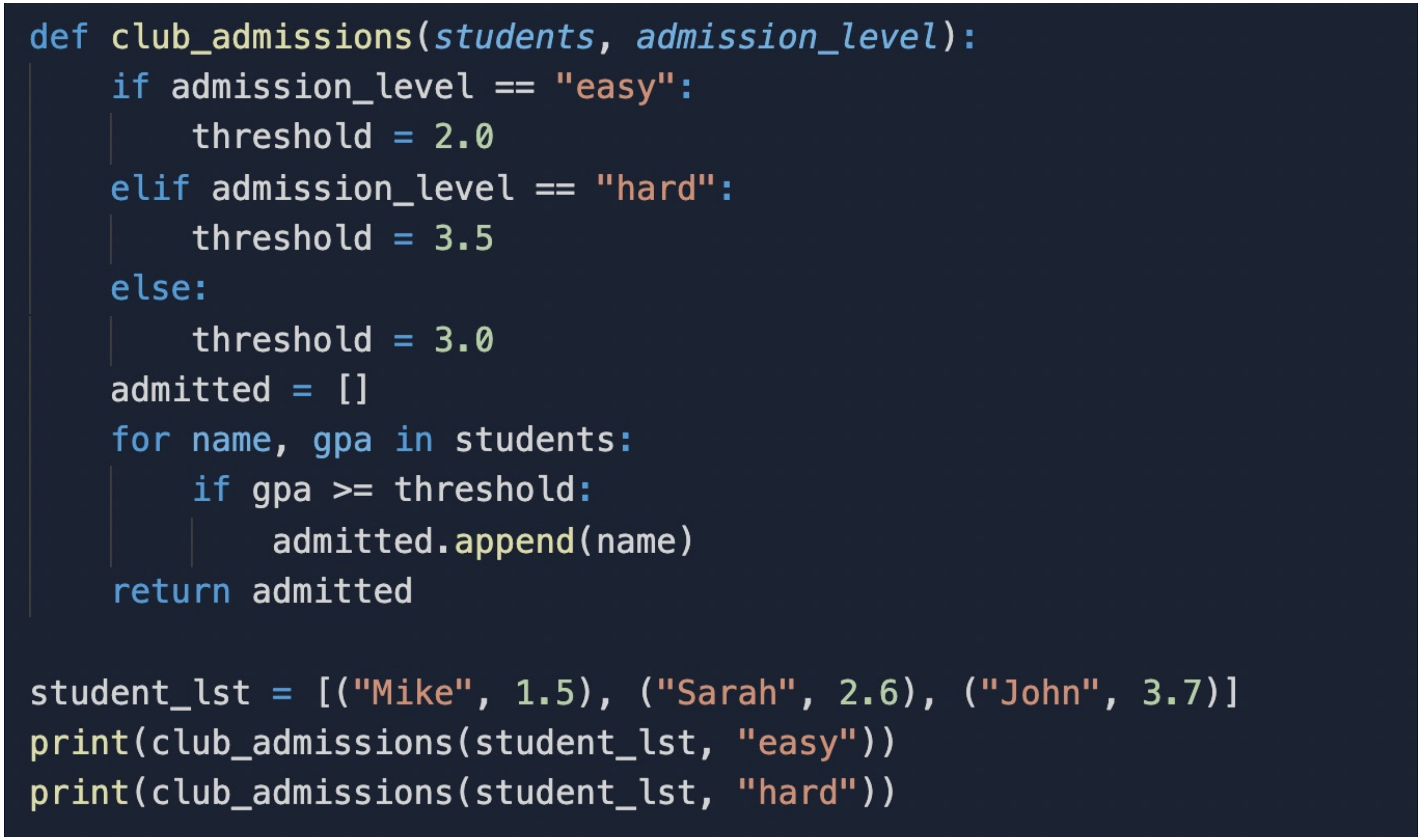
## Example 1 (Procedure example):



```python
def find_tutors(tutors, criteria):
    my_tutors = []
    if criteria == "price":
        for name, price, rating in tutors:
            if price <= 15:
                my_tutors.append(name)
    elif criteria == "rating":
        for name, price, rating in tutors:
            if rating >= 4.0:
                my_tutors.append(name)
    else:
        print("Invalid criteria.")
    return my_tutors

tutor_lst = [("Mike", 12, 3.5), ("Sarah", 18, 4.3), ("John", 24, 4.1)]
print(find_tutors(tutor_lst, "price"))
print(find_tutors(tutor_lst, "rating"))
```

## Example 2(Procedure example):



```python
def club_admissions(students, admission_level):
    if admission_level == "easy":
        threshold = 2.0
    elif admission_level == "hard":
        threshold = 3.5
    else:
        threshold = 3.0
    admitted = []
    for name, gpa in students:
        if gpa >= threshold:
            admitted.append(name)
    return admitted

student_lst = [("Mike", 1.5), ("Sarah", 2.6), ("John", 3.7)]
print(club_admissions(student_lst, "easy"))
print(club_admissions(student_lst, "hard"))
```

## Some Ideas

1) We did a lab on Alcohol Consumption. That dataset and many other interesting datasets can be found on Kaggle(https://www.kaggle.com/datasets). Find a dataset that is interesting and use it to do something similar like the Alchohol consumption lab. Datasets on Kaggle tend to include many columns; it might be necessary to delete most of the columns and keep only a few.
2) Kaggle datasets are in a nice csv file format. If you are looking for data elsewhere, you might need to convert your dataset to a text file. More examples: NFL quarterback statistics, NBA statistics, market capitalization or stock prices of companies, cars data, climate data, etc…
3) Write a game or app using Processing.