**Project:** Computer Vision-Based Roll Call System using Python

Task: Create a vision-based roll-call system.

Create a Development Environment

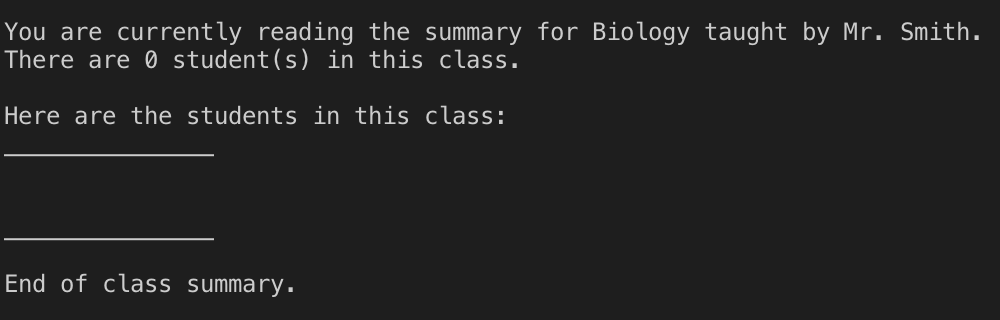
1. Open **Anaconda Navigator**.
2. Click **Environments**.
3. Create a new environment named “*rollcallsystem*” with Python 3.7.

Getting Started

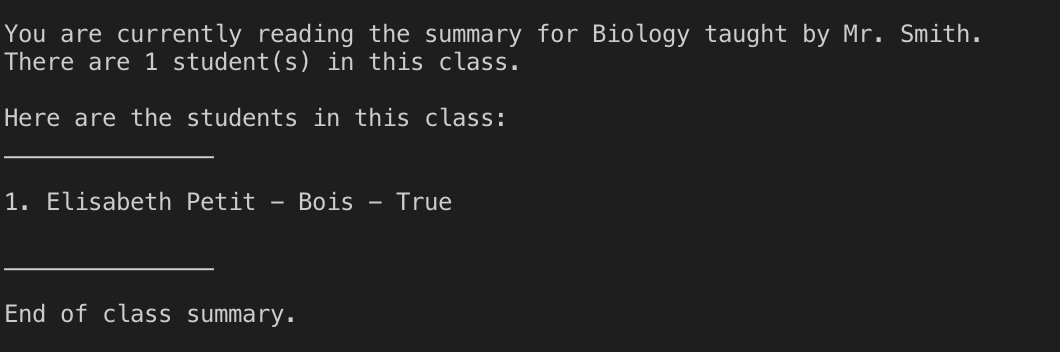
1. In a new Python file named “main.py,” create two classes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Initialization** | **Class Attributes** | **Class Functions** |
| Class | Takes in a *name* and *professor* | Name (string), professor (string), students (array) | getName – returns class name  setName – sets class name  getProfessor – returns class professor  setProfessor – sets class professor  getStudents – returns a list of students in a class  addStudent – adds a student to a class  removeStudent – removes a student from a class  printSummary – prints the class name, class professor, number of students, and a list of students |
| Student | Takes in a *name* | Name (string),  Absent (Boolean),  Sample (array) | getName – returns student name  setName – sets student name  isAbsent – returns if student is absent or not  set\_present – sets Absent attribute to false  set\_absent – sets Absent attribute to true |

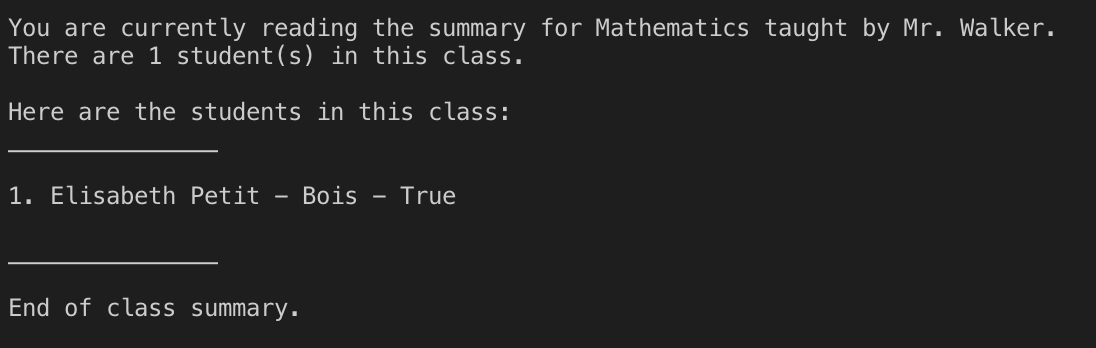
1. In main.py, define a *main* function that creates a “Biology” class with “Mr. Smith” as the professor.
2. Run the printSummary function on your new class. It should print the following:



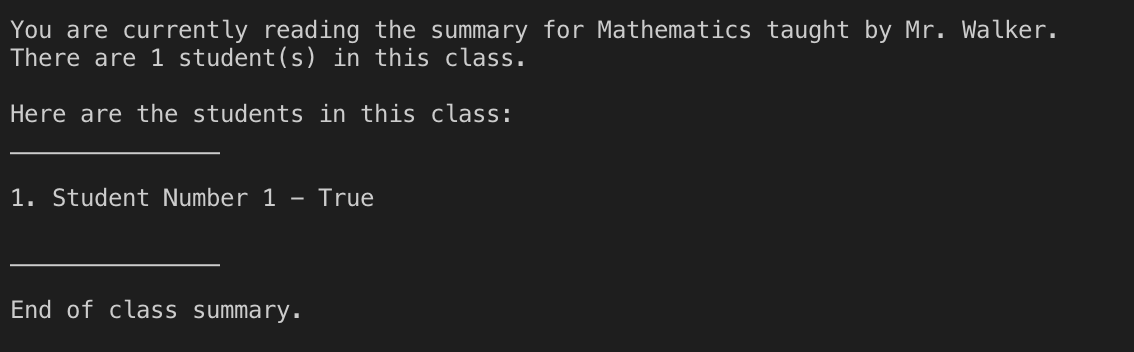
1. Create a student with your name and add it to the class.
2. Run the printSummary function on your class. It should print the following:



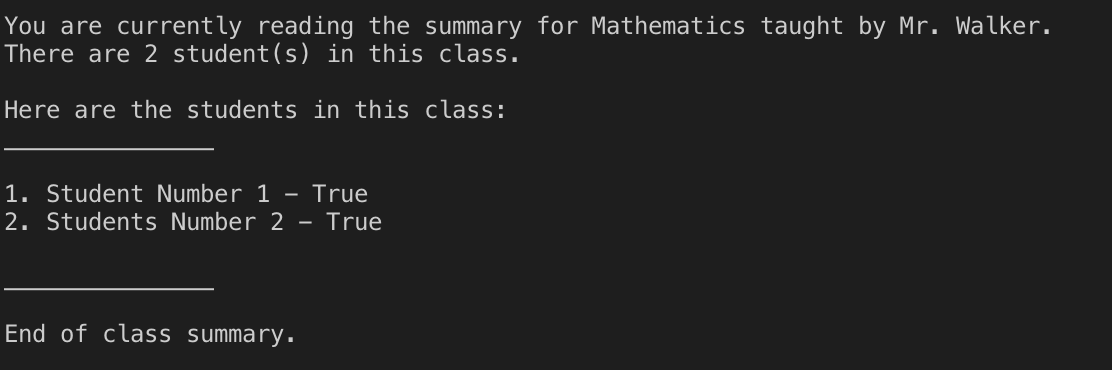
1. Change the name of the class to “Mathematics” and the professor to “Mr. Walker.”
2. Run the printSummary function on your class. It should print the following:



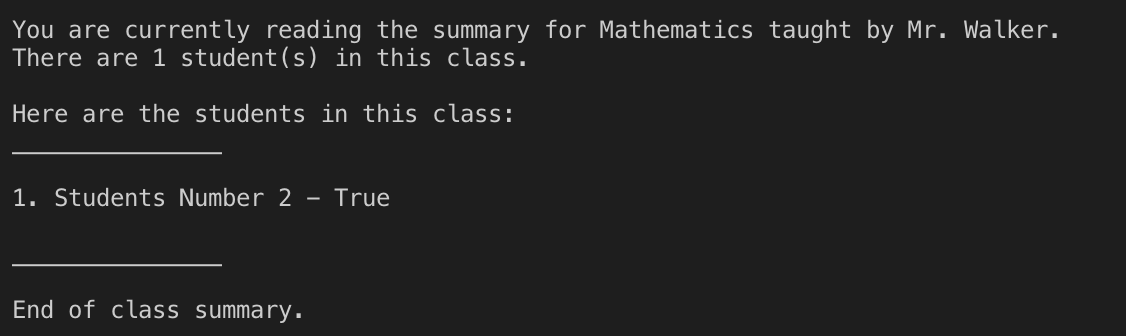
1. Change the name of the student in your class to “Student Number 1.”
2. Run the printSummary function on your new class. It should print the following:



1. Create “Student Number 2” and add it to the class.
2. Run the printSummary function on your class. It should print the following:

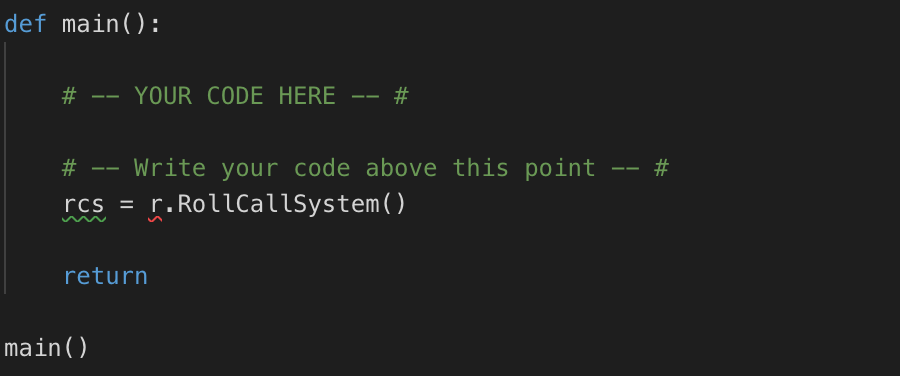


1. Remove “Student Number 1” from the class.
2. Run the printSummary function on your class. It should print the following:



Making the Roll Call System

1. Without deleting your previous code, modify your code to create a “Python 101” class taught by “Mr. Smith.” One of your group members should be the only student in the class.
2. In your “main” function, uncomment the line that reads “*rcs = r.RollCallSystem()*”

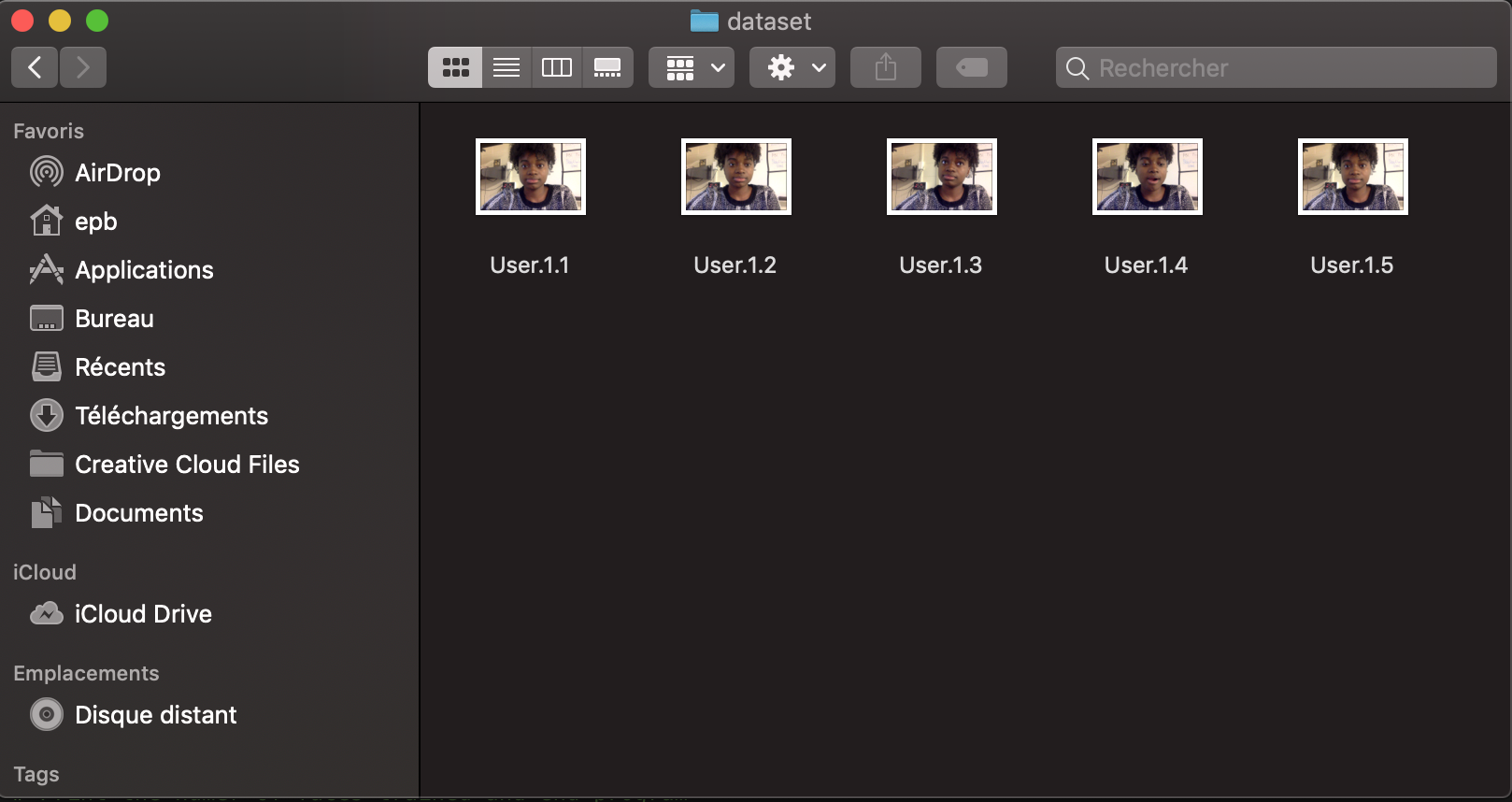


1. Pass the “Python 101” class object as a parameter for the r.RollCallSystem() function call.

Gathering Training Data

1. Using the webcam, take **FIVE** sample photos of **ONE** group member.
2. Save these photos in the project directory under the “dataset” folder.
3. Rename these photos using the following naming convention: User[#].[Photo#Here]. This name is meant to create a group member identifier and keep track of the photo number.

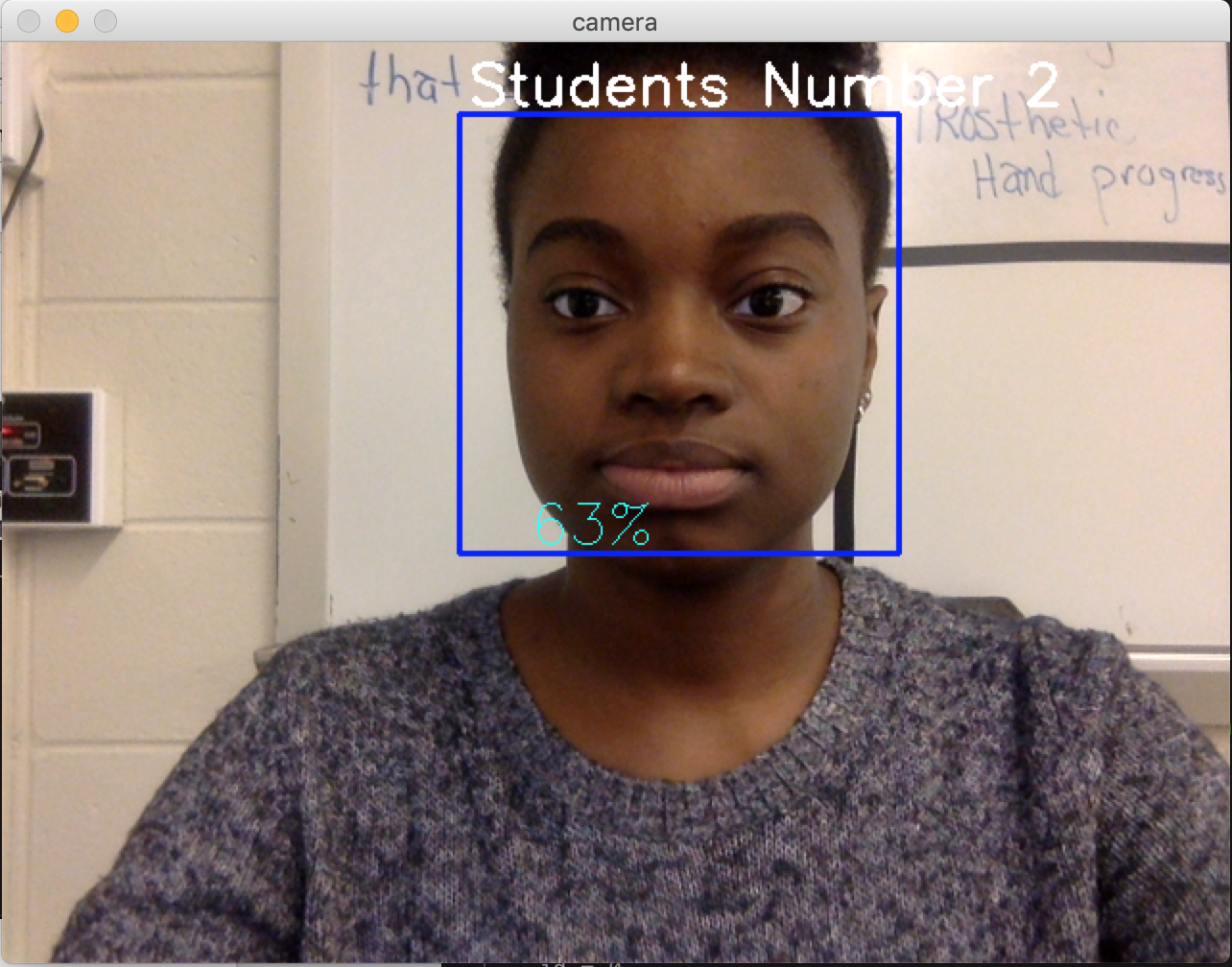
You should have 5 files in the “dataset” folder as illustrated below:



Testing Your Data

1. In the IDE terminal, run “source activate roll\_call”
2. Then, run “python main.py”
3. The program should print a summary of the class along with the absent status of the student.

A window should appear, showing the output of the webcam. The square identifies the group member as well as the confidence-level of your trained model.



Adding Students to Python 101

1. In “main.py,” modify your code to add **ALL** of your group members to “Python 101.” Remember the order you add everyone for later.
2. Print the class summary for the class object.

Gathering More Training Data

1. Using the webcam, take **FIVE** sample photos of **EACH** group member.
2. Save these photos in the project directory under the “dataset” folder.
3. Rename these photos using the following naming convention: User[#].[Photo#Here]. This name is meant to create a group member identifier and keep track of the photo number.

**<! -- The USER # should correspond with the ORDER that you added each group member. -- !>**

*If Stacey was added first in the code, her first photo should be named “User1.1”. If John was added second, his third photo should be named “User2.3”.*

By the time you finish, you should have (5 \* [Number of Members in Your Group]) files in the “dataset” folder.

1. Run the program using “python main.py”