|  |
| --- |
| Instructor |
| - instructors[]  - instructor\_passwords[] |
| + \_\_init\_\_(self, user\_name, password)  + addInstructor(self, user\_name)  + addInstructorPassword(self, password) |

UML Diagrams with two parent classes and their respective sub-class found in main.py

|  |
| --- |
| Instructor\_Name |
| +\_\_init\_\_(self, user\_name, password, first\_name, last\_name) |

|  |
| --- |
| Student |
| - students[]  - student\_passwords[] |
| + \_\_init\_\_(self, user\_name, password)  + addStudent(self, user\_name)  + addStudentPassword(self, password) |

|  |
| --- |
| Instructor\_Name |
| +\_\_init\_\_(self, user\_name, password, first\_name, last\_name) |

main.py

initialize lists students and studentpasswords, instructors, instructorpasswords

function checking if the instructor username already exists

if instructors list is empty

return False

else

for every instructor in the instructors list

if instructor is the same as username

return True

endfor

return False

function checking if the instructor password matches instructor username

index is set equal to the index of the username in instructors list

if instructorpasswords element at index is the same as password

display 'Login Successful'

else

while instructorpasswords element at index is not the same as password

input password

endwhile

display 'Login Successful

function checking if the student username already exists

if students list is empty

return False

else:

for every student in students lists:

if student is the same as username

return True

endfor

return False

function checking if the student password matches student username

index is set equal to the index of the username in students list

if studentpasswords element at index is the same as password

display 'Login Successful'

else

while instructorpasswords element at index is not the same as password

input password

endwhile

display 'Login Successful

create an object from the Instructor\_Name class

pass in the user entered username, password, firstname, lastname

pass in the username entered into the addInstructor method in Instructor class

pass in the password entered into the addInstructorPassword method in Instructor class

append the user entered username to instructors list

append the user entered password to instructor passwords list

create an object from the Instructor\_Name class

pass in the user entered username, password, firstname, lastname

pass in the username entered into the addInstructor method in Instructor class

pass in the password entered into the addInstructorPassword method in Instructor class

append the user entered username to instructors list

append the user entered password to instructor passwords list

create an object from the Instructor\_Name class

pass in the user entered username, password, firstname, lastname

pass in the username entered into the addInstructor method in Instructor class

pass in the password entered into the addInstructorPassword method in Instructor class

append the user entered username to instructors list

append the user entered password to instructor passwords list

create an object from the Student\_Name class

pass in the user entered username, password, firstname, lastname

pass in the username entered into the addStudent method in Student class

pass in the password entered into the addStudentPassword method in Student class

append the user entered username to students list

append the user entered password to studentpasswords list

create an object from the Student\_Name class

pass in the user entered username, password, firstname, lastname

pass in the username entered into the addStudent method in Student class

pass in the password entered into the addStudentPassword method in Student class

append the user entered username to students list

append the user entered password to studentpasswords list

create an object from the Student\_Name class

pass in the user entered username, password, firstname, lastname

pass in the username entered into the addStudent method in Student class

pass in the password entered into the addStudentPassword method in Student class

append the user entered username to students list

append the user entered password to studentpasswords list

begin

input user choice if they want to login or create an account

assert user entered choice 1 or 2

exception

assertion error

display 'Invalid Choice' if user does not pick choice 1 or 2

else

if user selects to login or create account, execute the statements below

if user selects to login

begin

input what user wants to sign in as an instructor or a student

assert user entered choice

exception

assertion error

display 'Invalid Choice' if user does not pick choice 1 or 2

else

if user selects to login as instructor or student, execute the statements below

if user selects to login as an instructor

display statement 'Instructor Login'

input get user to enter a username

call: function checking if the instructor username already exists

for every instructor in the list instructors

if the username exists in the instructor list

input get user to input the password

call: function checking if the instructor password matches instructor username

once logged in, import the module keylogger

break

else

input get user to re-enter username

call: function checking if the instructor username already exists

endfor

else if user selects to login as a student

display 'InstructorStudent Login’

input get user to enter a username

call: function checking if the student username already exists

for every instructor in the list instructors

if the username exists in the students list

input get user to input the password

call: function checking if the student password matches student username

once logged in, import the module keylogger

break

else:

input get user to re-enter username

call: function checking if the student username already exists

endfor

if user selects to create account

begin

input get user choice to create an account as an instructor or a student

assert user entered choice

exception

assertion error

display 'Invalid Choice' if user does not pick choice 1 or 2

else:

if user selects choice 1 or 2, execute the statements below

if user selects to create an account as an instructor

display statement 'Welcome Instructor'

input get user to enter a username

call: function checking if the instructor username already exists

for every instructor in instructors list

if the username exists in the instructors list

input get user to enter a different username

endfor

input get user to enter a password

input get user to enter firstname

input get user to enter lastname

create an object from the Instructor\_Name class

pass in the user entered username, password, firstname, lastname

pass in the username entered into the addInstructor method in Instructor class

pass in the password entered into the addInstructorPassword method in Instructor class

append the user entered username to instructors list

append the user entered password to instructor passwords list

else if user selects to create an account as a student

display statement 'Welcome Student'

input get user to enter a username

call: function checking if the student username already exists

for every student in the students list:

if the username exists in the students list

input get user to enter a different username

endfor

input get user to enter a password

input get user to enter firstname

input get user to enter lastname

create an object from the Student\_Name class

pass in the user entered username, password, firstname, lastname

pass in the username entered into the addStudent method in Student class

pass in the password entered into the addStudentPassword method in Student class

append the user entered username to students list

append the user entered password to studentpasswords list

module

keylogger.py

import Listener as Key from the pynput.keyboard library

a list named keys is initialized

a variable named count is initialized to zero

function captures keys that are pressed

keystrokes passed in as keys

keys and count set to global variables

keystrokes are appended to the list keys

count is incremented by one

if count is greater than or equal to 1

count is set to zero

call: write keys to a file function

list keys set to empty

function writes keys into a text file

pass in keys

opens the text file log.txt in append mode

for every key in keys list

variable k is assigned to key converted into strings and single quotation marks are eliminated

if space key is pressed

a ' ' (space) will be written to the file

close file

else if enter key is pressed

cursor will move to a new line in the text file

close file

else if other keys are without the word 'Key' are found/pressed

keys will be written into the file

close file

endfor

function captures keys that are released

if esc key is hit

program will stop running

Collect events until released, capture keys until released