

OOP_PCOM7E September 2022 Object Oriented Programming

Unit 4: Applying a UML Model to a Program Implementation: UML in Practice

Expand upon the activity diagram with the development of a class diagram using UML to support a system with basic employee-related functionality. This should include the retention of employee details and allowing an employee to book a day of annual leave.

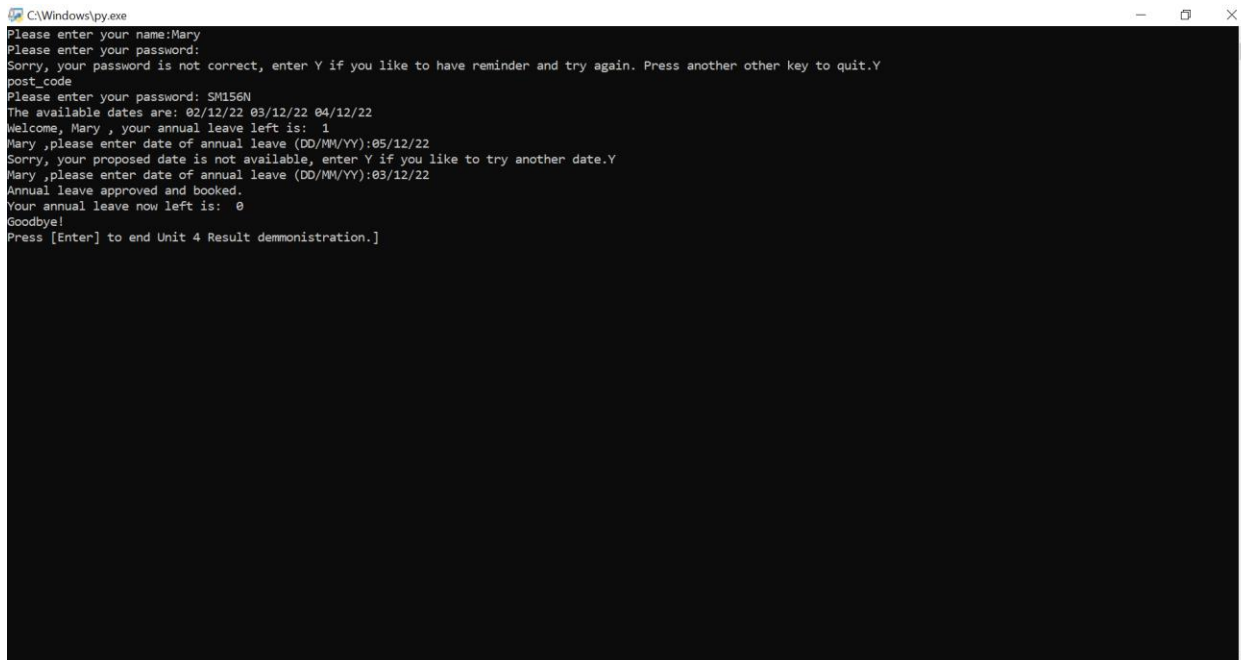
Develop the Python program to implement the class model.

Notes to readers:

This exercise requests me to write the class diagram and the activity diagram for the employee-related function from Unit 2. On top of a straight forward annual leave booking, the verification of user's identity is added. A user is first required to enter his/her name. In case the name does not match a registered user, an error message will come out. If the name is found as a registered user, password is required. In case the user forgets the password, it could call a "reminder" message.

The result of the code is set out below. The source code is set out a separate file. In case you are interested in it you could find it in the separate file.

Result



```
C:\Windows\py.exe
Please enter your name: Mary
Please enter your password:
Sorry, your password is not correct, enter Y if you like to have reminder and try again. Press another other key to quit.Y
post_code
Please enter your password: SM156N
The available dates are: 02/12/22 03/12/22 04/12/22
Welcome, Mary , your annual leave left is: 1
Mary ,please enter date of annual leave (DD/MM/YY):05/12/22
Sorry, your proposed date is not available, enter Y if you like to try another date.Y
Mary ,please enter date of annual leave (DD/MM/YY):03/12/22
Annual leave approved and booked.
Your annual leave now left is: 0
Goodbye!
Press [Enter] to end Unit 4 Result demmonstration.]
```

Program

creating class

class STAFF:

def __init__(self, name, age, post, salary):

self._name = name

self._age = age

self._post = post

self._salary = salary

class ANNUALCAL(STAFF):

def __init__(self, name, age, post, salary, no_leaveday_left,password,reminder):

STAFF.__init__(self, name, age, post, salary)

self._AL_left = no_leaveday_left

self._password = password

self._reminder = reminder

def print_AL_left(self):

AL = int(self._AL_left)

#Assume there are several days available for annual leave:

D1 = "02/12/22"

D2 = "03/12/22"

D3 = "04/12/22"

def PW(s):

Y = True

while Y == True:

password_enter = input("Please enter your password: ")

if password_enter == s._password:

Y = False

else:

PX = input("Sorry, your password is not correct, enter Y if you like to have reminder and try again.

Press another other key to quit.")

if PX == "Y" or DX == "y" or DX == "Yes" or DX == "yes":

print(s._reminder)

else:

print("Thank you for using annual leave booking system.")

Y = False

def ALS(s):

AL = int(s._AL_left,)

print("Welcome,", s._name, ", your annual leave left is: ", AL)

X = True

while X == True:

msg = s._name + " ,please enter date of annual leave (DD/MM/YY):"

DD = input(msg)

if DD == D1 or DD == D2 or DD == D3:

print ("Annual leave approved and booked.")

AL = AL-1

```

    print("Your annual leave now left is: ", AL)
    X = False
else:
    DX = input("Sorry, your proposed date is not available, enter Y if you like to try another date.")
    if DX == "Y" or DX == "y" or DX == "Yes" or DX == "yes":
        pass
    else:
        print("Thank you for using annual leave booking system.")
        X = False

P1 = ANNUALCAL("Mary",20,"Clerk",10000,1,"SM156N", "post_code")

Z = True
while Z == True:
    name_enter = input("Please enter your name:")
    if name_enter == P1._name:
        PW(P1)
        break
    else:
        print("Sorry, your name does not match our staff list, please enter the correct name.")

print("The available dates are:", D1, D2, D3)

ALS(P1)

print("Goodbye!")

```