The Employee Management Project

Tyler Chotikamars

INT 100 Fundamentals of Information Technology & Literacy

Amjad Alkilani

12/6/2020

Employee Management System Explanation

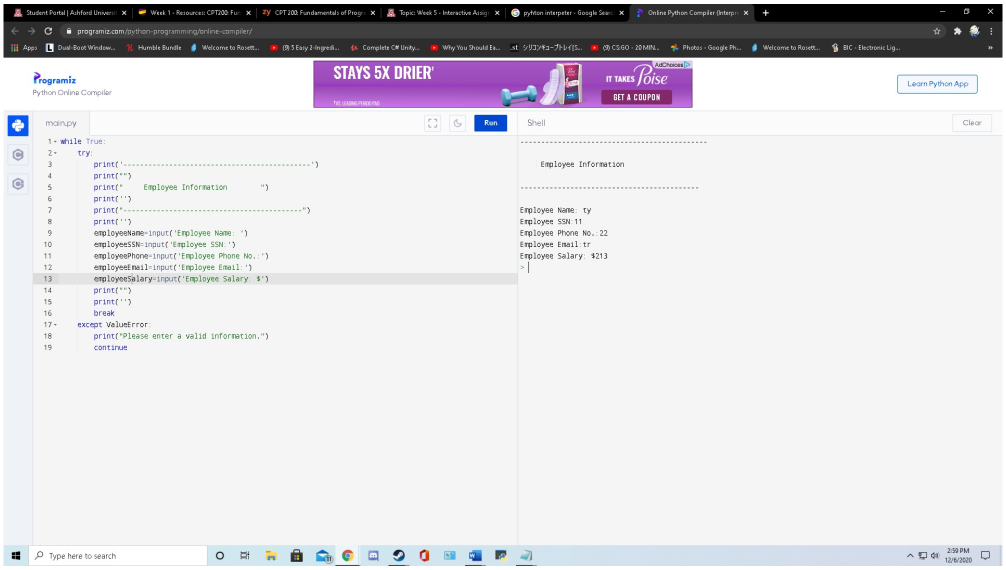
[no notes on this page]

Employee Management System Project

This project was fun, and I learned a lot through it! To combine everything into one single application I had to create a while True statement to collect information & have the program run continuously. For this program I defined functions that would be called when needed. I have a function for employee formatted information, one to view all employee information, one to add an employee, one to look them up by SSN, one to edit information, two for text files and finally a function that displays a main menu. I threw these functions into another while True statement so that depending on the users input, the program will carry out different tasks. With all my functions defined it was only a matter of putting them into order to get the desired output.

The purpose of this system is to collect a database of employee information. With this database you can import and export text files, it formats information, allows users to look employee's up by SSN, allows the user to edit information, allows the user to view either all or one employee's information, and add employee's.

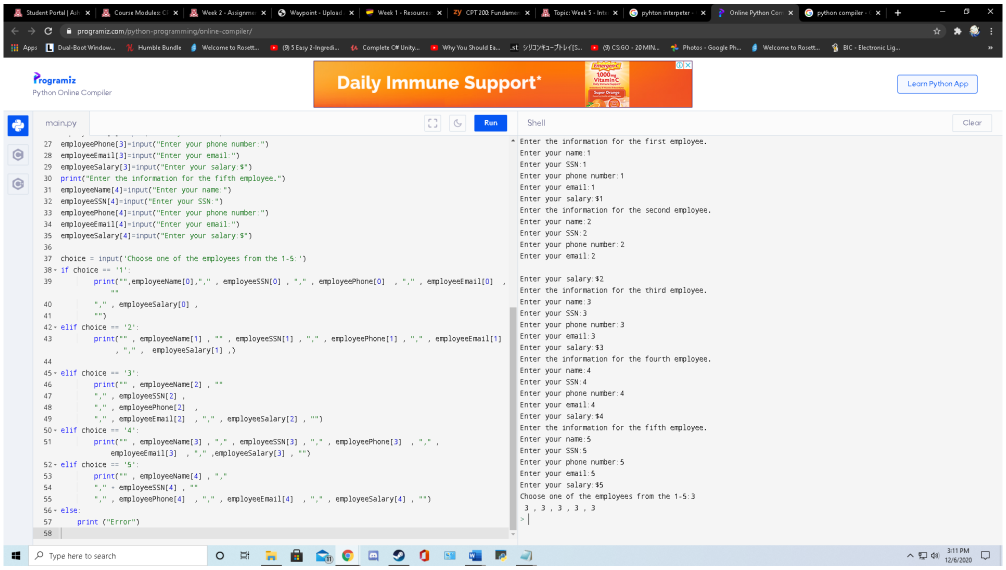
Functionality 1: This functionality is only meant to store the user input as variables such as employeeName and so on. It uses the users input to assign the variable.


Student Portal | Ashford Universit
Week 1 - Resources: CPT200: Fun
CPT 200: Fundamentals of Progra
Topic: Week 5 - Interactive Assign
pyhton interpeter - Google Searc
Online Python Compiler (Interpre
programiz.com/python-programming/online-compiler/
Programiz
Python Online Compiler
STAYS 5X DRIER
IT TAKES Poise
GET A COUPON
Learn Python App
main.py
Run
Shell
Clear
Type here to search
2:59 PM
12/6/2020

```
while True:
    try:
        print('------------------------------------------')
        print("")
        print("     Employee Information        ")
        print('')
        print("-----------------------------------------")
        print('')
        employeeName=input('Employee Name:  ')
        employeeSSN=input('Employee SSN:')
        employeePhone=input('Employee Phone No.:')
        employeeEmail=input('Employee Email:')
        employeeSalary=input('Employee Salary: $')
        print("")
        print('')
        break
    except ValueError:
        print("Please enter a valid information.")
        continue
```

```
------------------------------------------


    Employee Information


-----------------------------------------


Employee Name:  ty
Employee SSN:11
Employee Phone No.:22
Employee Email:tr
Employee Salary: $213
>
```

Functionality 2: This is code that I made to store 5 employee’s information while also displaying information from an employee the user chooses. It is designed to only print the employee the user selected.

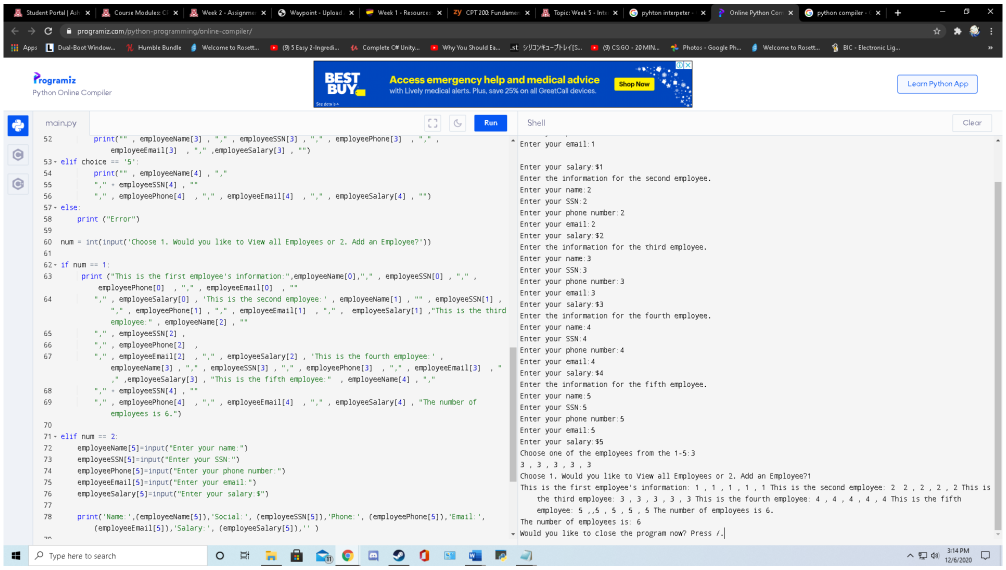


```
employeePhone[3]=input("Enter your phone number:")
employeeEmail[3]=input("Enter your email:")
employeeSalary[3]=input("Enter your salary:$")
print("Enter the information for the fifth employee.")
employeeName[4]=input("Enter your name:")
employeeSSN[4]=input("Enter your SSN:")
employeePhone[4]=input("Enter your phone number:")
employeeEmail[4]=input("Enter your email:")
employeeSalary[4]=input("Enter your salary:$")

choice = input('Choose one of the employees from the 1-5:')
if choice == '1':
        print("",employeeName[0],"," , employeeSSN[0] , "," , employeePhone[0]  , "," , employeeEmail[0]  ,
            ""
        "," , employeeSalary[0] ,
        "")
elif choice == '2':
        print("" , employeeName[1] , "" , employeeSSN[1] , "," , employeePhone[1] , "," , employeeEmail[1]
             , "," ,  employeeSalary[1] ,)

elif choice == '3':
        print("" , employeeName[2] , ""
        "," , employeeSSN[2] ,
        "," , employeePhone[2]  ,
        "," , employeeEmail[2]  , "," , employeeSalary[2] , "")
elif choice == '4':
        print("" , employeeName[3] , "," , employeeSSN[3] , "," , employeePhone[3]  , "," ,
            employeeEmail[3]  , "," ,employeeSalary[3] , "")
elif choice == '5':
        print("" , employeeName[4] , ","
        "," + employeeSSN[4] , ""
        "," , employeePhone[4]  , "," , employeeEmail[4]  , "," , employeeSalary[4] , "")
else:
    print ("Error")
```

```
Enter the information for the first employee.
Enter your name:1
Enter your SSN:1
Enter your phone number:1
Enter your email:1
Enter your salary:$1
Enter the information for the second employee.
Enter your name:2
Enter your SSN:2
Enter your phone number:2
Enter your email:2

Enter your salary:$2
Enter the information for the third employee.
Enter your name:3
Enter your SSN:3
Enter your phone number:3
Enter your email:3
Enter your salary:$3
Enter the information for the fourth employee.
Enter your name:4
Enter your SSN:4
Enter your phone number:4
Enter your email:4
Enter your salary:$4
Enter the information for the fifth employee.
Enter your name:5
Enter your SSN:5
Enter your phone number:5
Enter your email:5
Enter your salary:$5
Choose one of the employees from the 1-5:3
 3 , 3 , 3 , 3 , 3
>
```

Functionality 3: This is a screenshot of the code that I made that is supposed to allow the user to add an employee or view all employees. For my final project, I updated this code to the sample code I was given in my feedback. However, the code I made for this problem works. It displays all the employee’s information when prompted and allows the user to add another user.

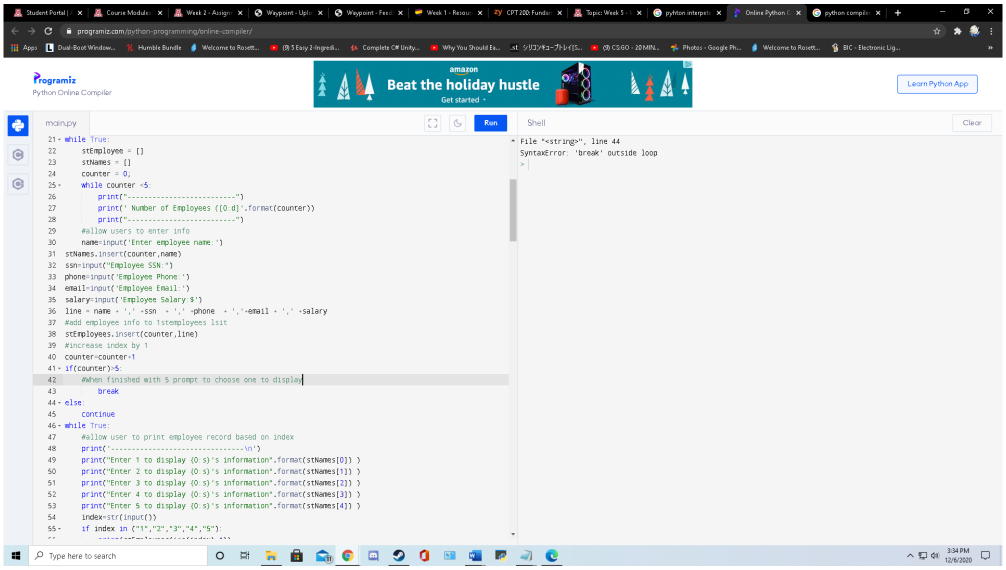
[no notes on this page]



```
    print("" , employeeName[3] , "," , employeeSSN[3] , "," , employeePhone[3]  , "," ,
        employeeEmail[3]  , "," ,employeeSalary[3] , "")
elif choice == '5':
    print("" , employeeName[4] , ","
    "," + employeeSSN[4] , ""
    "," , employeePhone[4]  , "," , employeeEmail[4]  , "," , employeeSalary[4] , "")
else:
  print ("Error")

num = int(input('Choose 1. Would you like to View all Employees or 2. Add an Employee?'))

if num == 1:
   print ("This is the first employee's information:",employeeName[0],"," , employeeSSN[0] , "," ,
      employeePhone[0]  , "," , employeeEmail[0]  , ""
    "," , employeeSalary[0] , 'This is the second employee:' , employeeName[1] , "" , employeeSSN[1] ,
      "," , employeePhone[1] , "," , employeeEmail[1]  , "," ,  employeeSalary[1] ,"This is the third
      employee:" , employeeName[2] , ""
    "," , employeeSSN[2] ,
    "," , employeePhone[2]  ,
    "," , employeeEmail[2]  , "," , employeeSalary[2] , 'This is the fourth employee:' ,
      employeeName[3] , "," , employeeSSN[3] , "," , employeePhone[3]  , "," , employeeEmail[3]  , "
      ," ,employeeSalary[3] , "This is the fifth employee:"  , employeeName[4] , ","
    "," + employeeSSN[4] , ""
    "," , employeePhone[4]  , "," , employeeEmail[4]  , "," , employeeSalary[4] , "The number of
      employees is 6.")

elif num == 2:
  employeeName[5]=input("Enter your name:")
  employeeSSN[5]=input("Enter your SSN:")
  employeePhone[5]=input("Enter your phone number:")
  employeeEmail[5]=input("Enter your email:")
  employeeSalary[5]=input("Enter your salary:$")

  print('Name:',(employeeName[5]),'Social:', (employeeSSN[5]),'Phone:', (employeePhone[5]),'Email:',
    (employeeEmail[5]),'Salary:', (employeeSalary[5]),'' )
```

```
Enter your email:1

Enter your salary:$1
Enter the information for the second employee.
Enter your name:2
Enter your SSN:2
Enter your phone number:2
Enter your email:2
Enter your salary:$2
Enter the information for the third employee.
Enter your name:3
Enter your SSN:3
Enter your phone number:3
Enter your email:3
Enter your salary:$3
Enter the information for the fourth employee.
Enter your name:4
Enter your SSN:4
Enter your phone number:4
Enter your email:4
Enter your salary:$4
Enter the information for the fifth employee.
Enter your name:5
Enter your SSN:5
Enter your phone number:5
Enter your email:5
Enter your salary:$5
Choose one of the employees from the 1-5:3
3 , 3 , 3 , 3 , 3
Choose 1. Would you like to View all Employees or 2. Add an Employee?1
This is the first employee's information: 1 , 1 , 1 , 1 , 1 This is the second employee: 2  2 , 2 , 2 , 2 This is
   the third employee: 3 , 3 , 3 , 3 , 3 This is the fourth employee: 4 , 4 , 4 , 4 , 4 This is the fifth
   employee: 5 ,,5 , 5 , 5 , 5 The number of employees is 6.
The number of employees is: 6
Would you like to close the program now? Press /.
```

Functionality 4: I could not get a screen shot of this function because my interpreter did not like the break in my count function. However, this is where I re-wrote all my code based on your feedback, which helped a lot! With your feedback I was able to code the functionalities for searching employees by their SSN and editing employee's information. However, after I re-wrote my code my interpreter could not run it, and I can't find any interpreter that will.

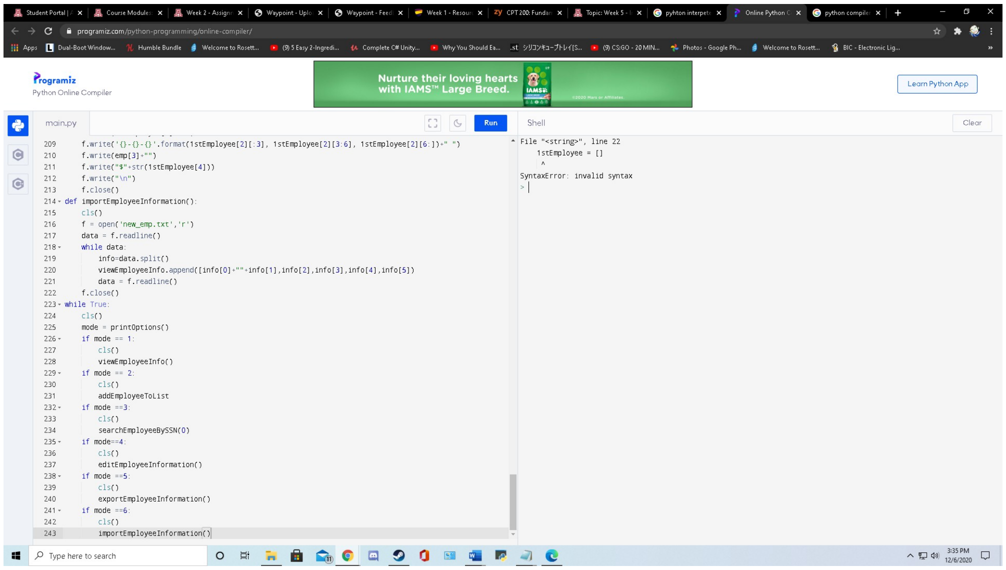


```
while True:
    stEmployee = []
    stNames = []
    counter = 0;
    while counter <5:
        print("------------------------")
        print(' Number of Employees ([0:d]'.format(counter))
        print("------------------------")
    #allow users to enter info
    name=input('Enter employee name:')
stNames.insert(counter,name)
ssn=input("Employee SSN:")
phone=input('Employee Phone:')
email=input('Employee Email:')
salary=input('Employee Salary:$')
line = name + ',' +ssn  + ',' +phone  + ','+email + ',' +salary
#add employee info to 1stemployees lsit
stEmployees.insert(counter,line)
#increase index by 1
counter=counter+1
if(counter)>5:
    #When finished with 5 prompt to choose one to display
        break
else:
    continue
while True:
    #allow user to print employee record based on index
    print('------------------------------\n')
    print("Enter 1 to display {0:s}'s information".format(stNames[0]) )
    print("Enter 2 to display {0:s}'s information".format(stNames[1]) )
    print("Enter 3 to display {0:s}'s information".format(stNames[2]) )
    print("Enter 4 to display {0:s}'s information".format(stNames[3]) )
    print("Enter 5 to display {0:s}'s information".format(stNames[4]) )
    index=str(input())
    if index in ("1","2","3","4","5"):
```

```
File "<string>", line 44
SyntaxError: 'break' outside loop
>
```

[no notes on this page]

Functionality 5: This screenshot is supposed to show how my code works while importing and exporting employee text files. Unfortunately, my code will not run through any interpreter I put it through for some reason. This was the best way that I found to perform these functionalities.



```
    f.write('{}-{}-{}'.format(1stEmployee[2][:3], 1stEmployee[2][3:6], 1stEmployee[2][6:])+" ")
    f.write(emp[3]+"")
    f.write("$"+str(1stEmployee[4]))
    f.write("\n")
    f.close()
def importEmployeeInformation():
    cls()
    f = open('new_emp.txt','r')
    data = f.readline()
    while data:
        info=data.split()
        viewEmployeeInfo.append([info[0]+""+info[1],info[2],info[3],info[4],info[5])
        data = f.readline()
    f.close()
while True:
    cls()
    mode = printOptions()
    if mode == 1:
        cls()
        viewEmployeeInfo()
    if mode == 2:
        cls()
        addEmployeeToList
    if mode ==3:
        cls()
        searchEmployeeBySSN(0)
    if mode==4:
        cls()
        editEmployeeInformation()
    if mode ==5:
        cls()
        exportEmployeeInformation()
    if mode ==6:
        cls()
        importEmployeeInformation()
```

```
File "<string>", line 22
    1stEmployee = []
     ^
SyntaxError: invalid syntax
>
```

This project was fun, and I learned a lot through it! To combine everything into one single application I had to create a while True statement to collect information & have the program run continuously. For this program I defined functions that would be called when needed. I have a function for employee formatted information, one to view all employee information, one to add an employee, one to look them up by SSN, one to edit information, two for text files and finally a function that displays a main menu. I threw these functions into another while True statement so that depending on the users input, the program will carry out different tasks. With all my functions defined it was only a matter of putting them into order to get the desired output.

The purpose of this system is to collect a database of employee information. With this database you can import and export text files, it formats information, allows users to look employee’s up by SSN, allows the user to edit information, allows the user to view either all or one employee’s information, and add employee’s.

```
while True:

    try:

        print('---------------------------------------------')

        print("")

        print("    Employee Information        ")
```

[no notes on this page]

```
        print(")
        print("-------------------------------------------")
        print(")
        name=input('Employee Name: ')
        ssn=input('Employee SSN:')
        phone=input('Employee Phone No.:')
        email=input('Employee Email:')
        salary=input('Employee Salary: $')
        print("")
        print(")
        break
    except ValueError:
        print("Please enter a valid information.")
        continue

while True:
    1stEmployee = []
    1stNames = []
    counter = 0;
    while counter <5:
        print("--------------------------")
        print(' Number of Employees ([0:d]'.format(counter))
        print("--------------------------")
    #allow users to enter info
    name=input('Enter employee name:')
1stNames.insert(counter,name)
ssn=input("Employee SSN:")
phone=input('Employee Phone:')
email=input('Employee Email:')
salary=input('Employee Salary:$')
```

```
line = name + ',' +ssn  + ',' +phone  + ','+email + ',' +salary
#add employee info to 1stemployees lsit
1stEmployees.insert(counter,line)
#increase index by 1
counter=counter+1
if(counter)>5:
    #When finished with 5 prompt to choose one to display
    break
else:
    continue
While True:
    #allow user to print employee record based on index
    print('--------------------------------\n')
    print("Enter 1 to display {0:s}'s information".format(1stNames[0]) )
    print("Enter 2 to display {0:s}'s information".format(1stNames[1]) )
    print("Enter 3 to display {0:s}'s information".format(1stNames[2]) )
    print("Enter 4 to display {0:s}'s information".format(1stNames[3]) )
    print("Enter 5 to display {0:s}'s information".format(1stNames[4]) )
    index=str(input())
    if index in ("1","2","3","4","5"):
        print(1stEmployees(int(index)-1))
    else:
        continue




employeeSSN = 0
employeeSSN = 0
#display employee info in format
```

[no notes on this page]

```
def employeeFormatedInfo (name,ssn,phone,email, salary):
    print("")
    print('----------{0:s}----------'.format(name))
    print('SSN: {0:s}'.format(ssn))
    print('Phone: {0:s}'.format(phone))
    print('Email: {0:s}'.format(email))
    print('Salary: ${0:s}'.format(salary))
    print("--------------------")
    print("")
#view all employees
def viewEmployeeInfo ():
    cls()
    print("")
    print("------------------------------")
    print('     View all employees in the system')
    print("")
    print("------------------------------")
    print("")
    if(len(1stEmployee)==0):
        print("No employees in the list.\n")
    else:
        for i in range(0, len(1stEmployee)):
            line = 1stEmployee(i).split(',')
            employeeFormatedInfo (line[0], line[1], line[3], line [4])
    try:
        option=int(input('Enter 0 to exit or any key to continue:'))
        if option == 0:
            cls()
    except:
        cls()
```

[no notes on this page]

```
    input('Press any key to go back to the Main Menu:')
    cls()
#add employee
def addEmployeeToList():
    cls()
    print("-----------------------------------")
    print("")
    print("Add Employee Information")
    print("")
    print("-----------------------------------")
    print("")
    try:
        name=input("Employee Name: ")
        ssn=input("Employee SSN:")
        phone=input("Employee Phone:")
        email=input("Employee Email:")
        salaryinput("Employee Salary: $")
        line = name +','+ssn+','+phone+','+email+','+salary
        index = len(1stEmployee)
        1stEmployee.insert(indext, line)
    except:
        cls()
        addEmployeetoList()
    print("")
    print("\nEmployee information has been successfully added to the list.\n")
    print("")
    print("---------------------------------")
    print("")
    try:
        #allow to add users or main menu
        option=input('Enter q/Q to return to main menu, or any other key to add more employees:')
        if option.lower() =="q":
            cls()
        else:
            cls()
            addEmployeeToList()
    except:
            cls()
            addEmployeeToList()
#display main menu
def printOptions():
    print("----------Employee Management System----------\n")
    print('There are ( {0:2d} ) employees in the system.'.format(len(stEmployee)))
    print("\n-----------------------------------")
    print('1. View all employees \n')
    print('2. Add new employee \n')
    print('3.Search employee by SSN \n')
    print('4. Edit employee information \n')
    #validate user selection
    try:answer=int(input('Please enter your option number: '))
    except ValueError:
        print("Not a number")
        return 100
    print("")
    print("---------------------------------")
    print("")
    return answer
def searchEmployeeBySSN (isForEdit):
    cls()
    print("Search Employee by SSN")
    if(len(stEmployee)==0):
        input("\n   No employees in the list.\n")
        cls()
    else:
        try:
            ssn=input('Please enter q/Q to exit or enter a new employee SSN: ')
            global employeeSSN
            employeeSSN = ssn
            if ssn.lower() =="q":
                return 0
        except ValueError:
            searchEmployeeBySSN(0)
    for i in range(0, len(stEmployee)):
        line = stEmployee(i).split(',')
        if(line[1] == ssn):
            global employeeIndex
            employeeIndex = 1
            employeeFormatedInfo (line[0],line[1], line[2], line[3], line[4])
            break
        else:
            print("\n   Can't find an employee with the SSN you provided.\n")
            return 0
try:
    if(isForEdit==0):
        option=input('Enter q/Q to exit or any key to search for another employee:')
        if option.lower() =="q":
            cls()
        else:
            searchEmployeeBySSN(0)
except:
```

[no notes on this page]

```
    cls()


def editEmployeeInformation():
cls()
#reuse the searchEmployeeBySSN
result = searchEmployeeBySSN(1)
if(result != 0) :
    name=input("Employee new name: ")
    phone=input("Employee new phone no.:")
    email=input("Employee new email:")
    salary=input("Employee new salary: $")
    #delete old info for list
    del 1stEmployee[employeeIndex]
    line = name + ',' +employeeSSN + ',' +phone + ',' +email + ','+salary
    #add new info
    1stEmployee.insert(employeeIndex, line)
    input("\nEmployee information has successfully been updated. Please enter an key to return to the Main Menu.")
def exportEmployeeInformation():
    cls()
    f = open('export.txt', 'w')
    for emp in viewEmployeeInfo:
            f.write(1stEmployee[0].title()+"")
            f.write(1stEmployee[1]+" ")
            f.write('{}-{}-{}'.format(1stEmployee[2][:3], 1stEmployee[2][3:6], 1stEmployee[2][6:])+" ")
            f.write(emp[3]+"")
            f.write("$"+str(1stEmployee[4]))
            f.write("\n")
    f.close()
def importEmployeeInformation():
```

[no notes on this page]

```
    cls()
    f = open('new_emp.txt','r')
            data = f.readline()
            while data:
                    info=data.split()
                    viewEmployeeInfo.append([info[0]+""+info[1],info[2],info[3],info[4],info[5])
                    data = f.readline()
    f.close()
while True:
    cls()
    mode = printOptions()
    if mode == 1:
        cls()
        viewEmployeeInfo()
    if mode == 2:
        cls()
        addEmployeeToList
    if mode ==3:
        cls()
        searchEmployeeBySSN(0)
    if mode==4:
        cls()
        editEmployeeInformation()
    if mode ==5:
        cls()
        exportEmployeeInformation()
    if mode ==6:
        cls()
        importEmployeeInformation()
```

References:

Miller, B., Vahid, F., & Lysecky, R. (2015). *Programming in Python 3*. Retrieved from https://zybooks.zyante.com/

[no notes on this page]