./

Advanced Python

Submitted by:

99003163



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1.0 | 13-12-2020 | SHREYA KORI |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

# ADVANCED PYTHON PROGRAMMING

## **ACTIVITY 1:**

## **CONTACT BOOK:**

CODES (IMPLEMENTATION USING CLASS AND METHODS):

class Contact\_book: # CONTACT BOOK

print("CONTACT BOOK")

def option\_entry(): # Defines the options for phone book

option = int(input("""Which of the following operations would you like to perform\n

1. Display contacts\n

2. Add contact\n

3. Check contact\n

4. Delete contact\n

5. Exit\n"""))

return option # Returns the entered option for further process

def contact\_no(): # Function to check validity of a number

number = int(input("Enter number to check validity:\n"))

n = number

count = 0

while(n > 0):

count = count+1

n = n//10

return count

def phone\_book(): # Defines a Phone book

phonebook = {} # Empty phone book to store values

while True: # Initiating a loop to run the program

opt = option\_entry()

if opt == 1: # Condition to display the contacts already present

if bool(phonebook) is True:

for i, j in phonebook.items():

print(i, '-->', j)

else: # Condition to display if phone book is empty

print("Phonebook is empty\n")

elif opt == 2: # Condition to add a new contact

no\_of\_digits = contact\_no() # call to check validity

if no\_of\_digits == 10:

print("Number is Valid")

phone\_no = input("Enter a number for entry\n")

person\_name = input("Enter a name for entry\n")

if person\_name not in phonebook: # Condition to update contact

phonebook.update({person\_name: phone\_no})

print("Contact has been successfully saved\n")

else:

print("Contact already exists\n")

elif opt == 3: # Condition to check contacts

name = input("Enter the name whose contact you want to view\n")

if name in phonebook:

print("Contact already exists\n")

else:

print ("Go to the option entry to add new contact\n")

elif opt == 4: # Condition to Delete contacts

name = input("Enter the name whose contact you want to delete\n")

if name in phonebook:

print('contact', name, ':', phonebook[name])

confirmation = input("Are you sure you want to delete? Y/N\n")

if confirmation.capitalize == "Y": # Initiate a decision

phonebook.pop(name, None) # If yes then remove contact

# To print the rest of contacts in the phone book

for i, j in phonebook.items():

print(i, '-->', j)

else:

print("Return to option\_entry\n")

else:

print("Contact doesn't exist\n")

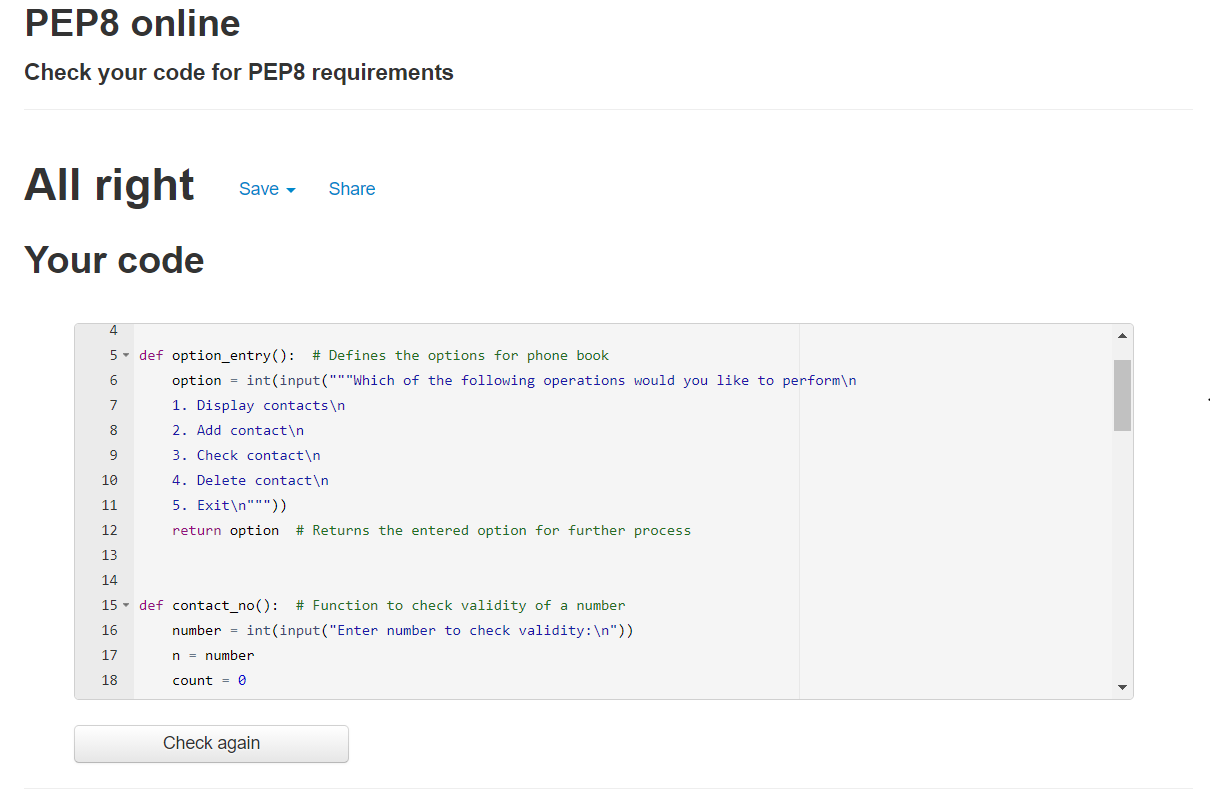
elif opt == 5: # condition to exit

print("THANK YOU\n")

break # Ends program

phone\_book()

**SCREENSHOT:**



## **CODES (IMPLEMENTATION OF OBJECT, SINGLE AND MULTIPLE INHERITANCE):**

class Contact\_book: # CONTACT BOOK

print("CONTACT BOOK")

def option\_entry(self): # Defines the options for phone book

option = int(input("""Which of the following operations would you like to perform\n

1. Display contacts\n

2. Add contact\n

3. Check contact\n

4. Delete contact\n

5. Exit\n"""))

return option # Returns the entered option for further process

class Validity:

def contact\_no(self): # Function to check validity of a number

fr1 = open("entry.txt", "r")

for each\_line in fr1:

number = int(each\_line[15:])

n = number

count = 0

while(n > 0):

count = count + 1

n = n//10

return count

fr1.close()

class Phone\_book(Contact\_book, Validity):

def phone\_book(self): # Defines a Phone book

phonebook = {} # Empty phone book to store values

while True: # Initiating a loop to run the program

opt = pb.option\_entry()

if opt == 1: # Condition to display the contacts already present

if bool(phonebook) is True:

for i, j in phonebook.items():

print(i, '-->', j)

else: # Condition to display if phone book is empty

print("Phonebook is empty\n")

elif opt == 2: # Condition to add a new contact

no\_of\_digits = pb.contact\_no() # call to check validity

print(no\_of\_digits)

if no\_of\_digits == 10:

print("Number is Valid")

fr1 = open("entry.txt", "r")

for each\_line in fr1:

phone\_no = each\_line[15:]

person\_name = each\_line[0:14]

if person\_name not in phonebook.keys():

# Condition to update contact

phonebook.update({person\_name: phone\_no})

print("Contact has been successfully saved\n")

elif person\_name in phonebook.keys():

print("Contact already exists\n")

fr1.close()

elif opt == 3: # Condition to check contacts

name = input("Enter the name whose contact you want to view\n")

if person\_name in phonebook.keys():

print("Contact already exists\n")

else:

print ("Go to the option entry to add new contact\n")

elif opt == 4: # Condition to Delete contacts

name = input("Enter the name whose contact you want to del\n")

if person\_name in phonebook.keys():

print('contact', person\_name, ':', phonebook[person\_name])

confirmation = input("Are you sure you want to del? Y/N\n")

if confirmation.capitalize == "Y": # Initiate a decision

phonebook.pop(name, None) # If yes then remove contact

# To print the rest of contacts in the phone book

for i, j in phonebook.items():

print(i, '-->', j)

else:

print("Return to option\_entry\n")

else:

print("Contact doesn't exist\n")

elif opt == 5: # condition to exit

print("THANK YOU\n")

break # Ends program

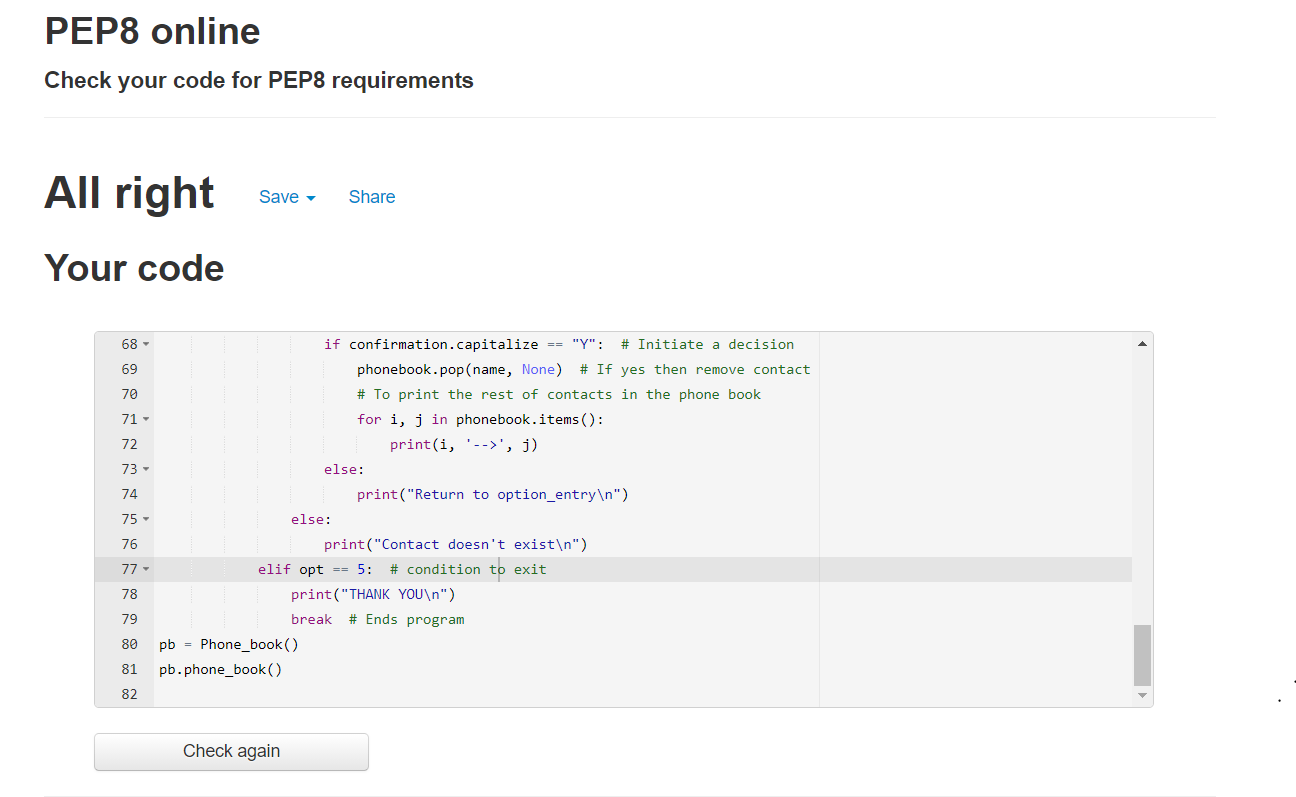
pb = Phone\_book()

pb.phone\_book()

INPUT FILE(TO READ):



**SCREENSHOT:**



## **ACTIVITY 2:**

## **IMPLEMENTATION OF REGULAR EXPRESSION:**

import re

ids = 'shrey98@google.com, kori@hotmail.com, shreyakori@yahoocom'

email = re.findall(r'[\w\.-]+@[\w\.-]+\.[a-z]{3}',ids)

for mail in email:

print(mail)

spl=re.split(r'\s',ids)

print(spl)

for id1 in spl:

print(re.match("[a-z 0-9]+@[a-z]+\.[a-z]{3}",id1))

**OUTPUT:**

