./

GENESIS- Advanced Python Programming Summary Report



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

**Details**

Contents

[Initial code: 4](#_Toc58752083)

[Source Code with OOPS concept: 7](#_Toc58752084)

[Regular Expression examples: 13](#_Toc58752085)

**Medical Store Management system**

# Initial code:

class pharmacy:  
    def \_\_init\_\_(self,dict, name):  
        self.medlist=dict  
        [self.name](http://self.name/)=name  
        self.cart={}  
    def displaymed(self):  
        print(f"we have following medicines: {[self.name](http://self.name/)}")  
        for med in self.medlist:  
            print(med)  
    def addmedtobuy(self,user,med,quan):  
        if med not in self.cart.keys():  
            self.cart.update({med:user})  
            print("cost of medicine",self.medlist[med])  
            #print(f"price of {med} is {self.medlist[med]}")  
            a=self.medlist[med]  
            b=a\*int(quan)  
            print(b)  
        else:  
            print(f"{self.cart[med]} is out of stock")  
    def addmed(self, med,cost):  
        if med not in self.medlist:  
            self.medlist.update({med:cost})  
            print(f"{med} is added to the medlist")  
    def deletemedicine(self,med):  
        cost=self.medlist[med]  
        self.medlist.pop((med))  
        print(f"{med} is deleted from the medlist")  
     
def main():  
    LTTS = pharmacy({"Crocin": 10, 'Cypon':20, 'alday':2, 'Histocart-b':4, 'coldact':5, 'Dolo':6}, "LTTS")  
  
    while(True):  
        print(f"Welcome to the {LTTS.name} pharmacy. Enter your choice to continue")  
        print("1. Display medicine")  
        print("2. add med to buy")  
        print("3. Add medine")  
        print("4. Delete medicine")  
        user\_choice = input()  
        if user\_choice not in ['1','2','3','4']:  
            print("Please enter a valid option")  
            continue  
  
        else:  
            user\_choice = int(user\_choice)  
  
  
        if user\_choice == 1:  
            LTTS.displaymed()  
  
        elif user\_choice == 2:  
            med = input("Enter the name of the med you want to buy:")  
            user = input("Enter your name")  
            quan = input("Enter the quantity")  
            LTTS.addmedtobuy(user, med, quan)  
  
        elif user\_choice == 3:  
            med = input("Enter the name of the med you want to add:")  
            cost=input("Enter the price of medicine:")  
            LTTS.addmed(med,cost)  
  
        elif user\_choice == 4:  
            med = input("Enter the name of the medicine you want to delete:")  
            LTTS.deletemedicine(med)  
  
        else:  
            print("Not a valid option")  
        print("Press q to quit and c to continue")  
        user\_choice2 = ""  
        while(user\_choice2!="c" and user\_choice2!="q"):  
            user\_choice2 = input()  
            if user\_choice2 == "q":  
                exit()  
  
            elif user\_choice2 == "c":  
                continue  
main()

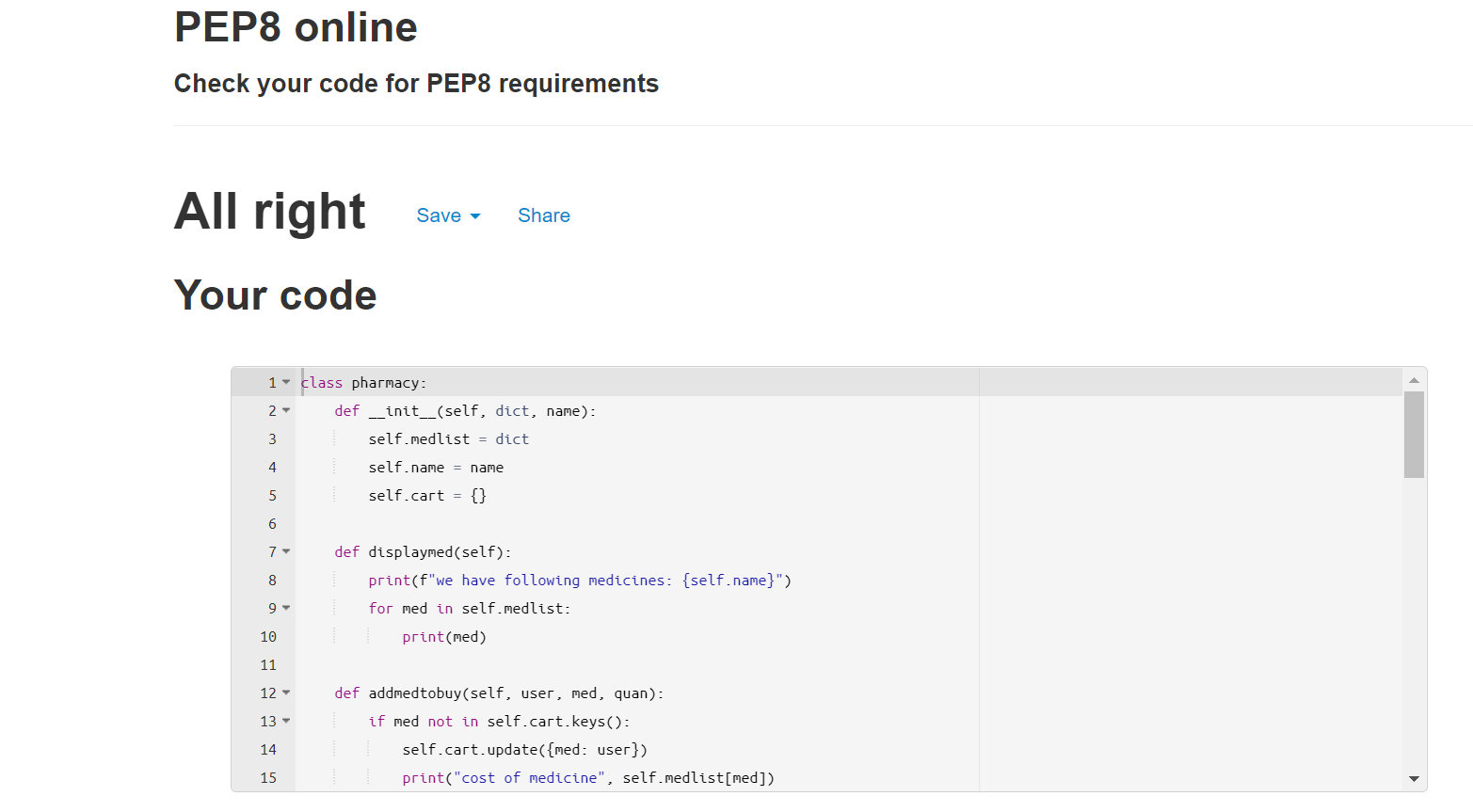


Figure 1 Code Quality

GITHUB LINK: <https://github.com/Genesis99003160/Python-Programming-Project>

## **Source Code with OOPS concept:**

import os

import sys

class medicalstore:

def \_\_init\_\_(self,dict,name):

medicalstore.medicinelist=dict

self.name=name

medicalstore.cart={}

class user(medicalstore):

def \_\_init\_\_(self,dict,name):

medicalstore.\_\_init\_\_(self,dict,name)

def displaymedicines(self):

print(f"we have following medicines: {self.name}")

for medicine in medicalstore.medicinelist:

print(medicine)

def purchasemedicine(self,user,medicine,quantity):

if medicine not in self.cart.keys():

medicalstore.cart.update({medicine:user})

print("cost of medicine",medicalstore.medicinelist[medicine])

value=medicalstore.medicinelist[medicine]

amount=value\*int(quantity)

print("amount to be paid",amount)

else:

print(f"{medicalstore.medicinelist[medicine]} is out of stock")

class admin(user):

def \_\_init\_\_(self,dict,name):

user.\_\_init\_\_(self,dict,name)

def addmedicinetostore(self, medicine,cost):

if medicine not in medicalstore.medicinelist:

medicalstore.medicinelist.update({medicine:cost})

print(f"{medicine} is added to the medlist")

def deletemedicinefromstore(self,medicine):

cost=medicalstore.medicinelist[medicine]

medicalstore.medicinelist.pop((medicine))

print(f"{medicine} is deleted from the medlist")

def main():

LTTS =admin({"Crocin": 10, 'Cypon':20, 'alday':2, 'Histocart-b':4, 'coldact':5, 'Dolo':6}, "LTTS")

#LTTS=admin()

print(os.name)

print(sys.path)

print(sys.version)

while(True):

admin\_user\_mode=input("Enter your mode")

if admin\_user\_mode=="1":

password=input("Enter your password")

if password=="1423":

while(True):

print(f"Welcome to the {LTTS.name} pharmacy. Enter your choice to continue")

print("1. Add medicine to store")

print("2. Delete medicine from store")

user\_choice = input()

if user\_choice not in ['1','2']:

print("Please enter a valid option")

continue

else:

user\_choice = int(user\_choice)

if user\_choice == 1:

medicine = input("Enter the name of the medicine you want to add:")

cost=int(input("Enter the price of medicine:"))

LTTS.addmedicinetostore(medicine,cost)

medicalstore.medicinelist.update({medicine:cost})

LTTS.displaymedicines()

elif user\_choice == 2:

medicine = input("Enter the name of the medicine you want to delete:")

LTTS.deletemedicinefromstore(medicine)

else:

print("Not a valid option")

print("Press q to quit and c to continue")

user\_choice2 = ""

#while(user\_choice2!="c" and user\_choice2!="q"):

user\_choice2 = input()

if user\_choice2 == "q":

break

#os.\_exit(0)

elif user\_choice2 == "c":

continue

else:

print("Incorrect Password")

elif admin\_user\_mode=="2":

password=input("Enter your password")

if password=="1213":

while(True):

print(f"Welcome to the {LTTS.name} pharmacy. Enter your choice to continue")

print("1. Display medicines")

print("2. add medicicne to purchase")

user\_choice = input()

if user\_choice not in ['1','2']:

print("Please enter a valid option")

continue

else:

user\_choice = int(user\_choice)

if user\_choice == 1:

LTTS.displaymedicines()

elif user\_choice == 2:

medicine = input("Enter the name of the medicine you want to buy:")

user = input("Enter the customer name")

quantity = input("Enter the quantity")

LTTS.purchasemedicine(user, medicine, quantity)

else:

print("Not a valid option")

print("Press q to quit and c to continue")

user\_choice2 = ""

#while(user\_choice2!="c" and user\_choice2!="q"):

user\_choice2 = input()

if user\_choice2 == "q":

break

#os.\_exit(0)

elif user\_choice2 == "c":

continue

else:

print("Incorrect Password")

else:

print("Invalid Mode")

main()

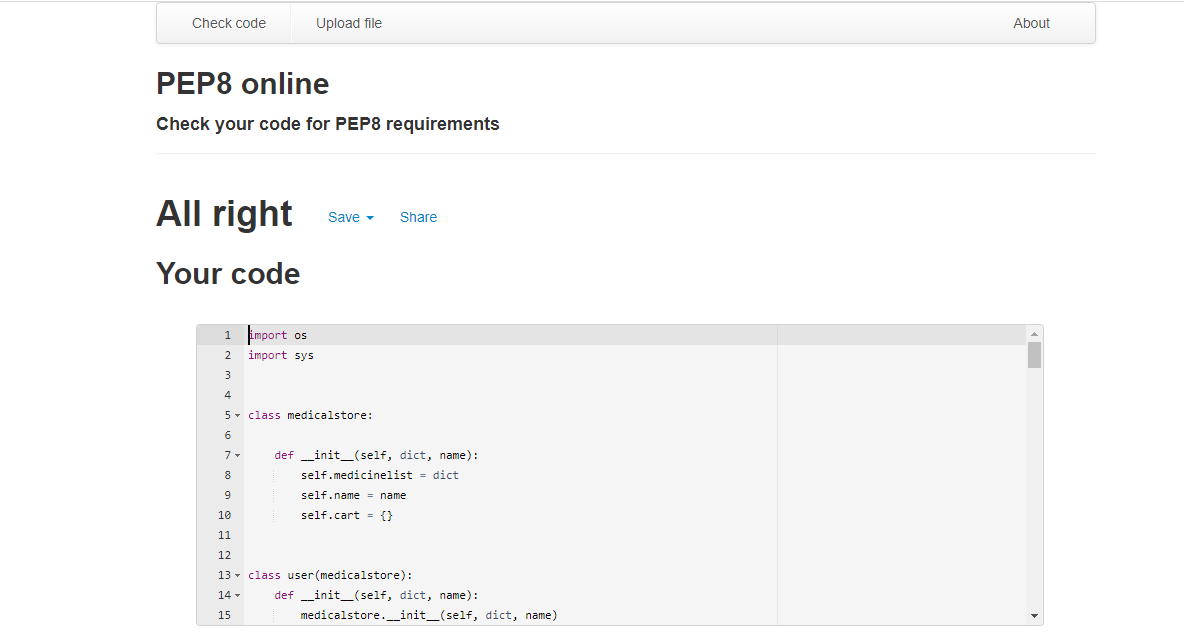


Figure 2 Code Quality with OOPS Concept

### REGULAR EXPRESSION EXAMPLES:

**Regular Expression 1:**

import re

print(re.findall("[a-z]+://","https://outlook.office.com.mcas.ms/mail/inbox"))

**Result:** <re.Match object; span=(0, 25), match='samudrala.apurva@ltts.com'>

**Regular expression 2:**

print(re.search("[a-z]+[A-Z]","samuDrala.apurva"))

**Result:** <re.Match object; span=(0, 5), match='samuD'>

**Regular Expression 3:**

print(re.search("[a-z]+://[a-z]+.[a-z]+.[a-z]{3}","https://www.youtube.co"))

      **Result:** <re.Match object; span=(0, 19), match='https://www.youtube'>

**Regular Expression 4:**

print(re.search("[a-z]+://[a-z]+.[a-z]+.[a-z]{3}","https://www.youtube.com"))

**Result:** <re.Match object; span=(0, 23), match='https://www.youtube.com'>

**Regular Expression 5:**

print(re.findall("[a-z]+://","https://outlook.office.com.mcas.ms/mail/inbox"))

**Result:** ['https://']

**Regular Expression 6:**

print(re.findall("[a-z]+","https://outlook.office.com.mcas.ms/mail/inbox"))

**Result:** ['https', 'outlook', 'office', 'com', 'mcas', 'ms', 'mail', 'inbox']

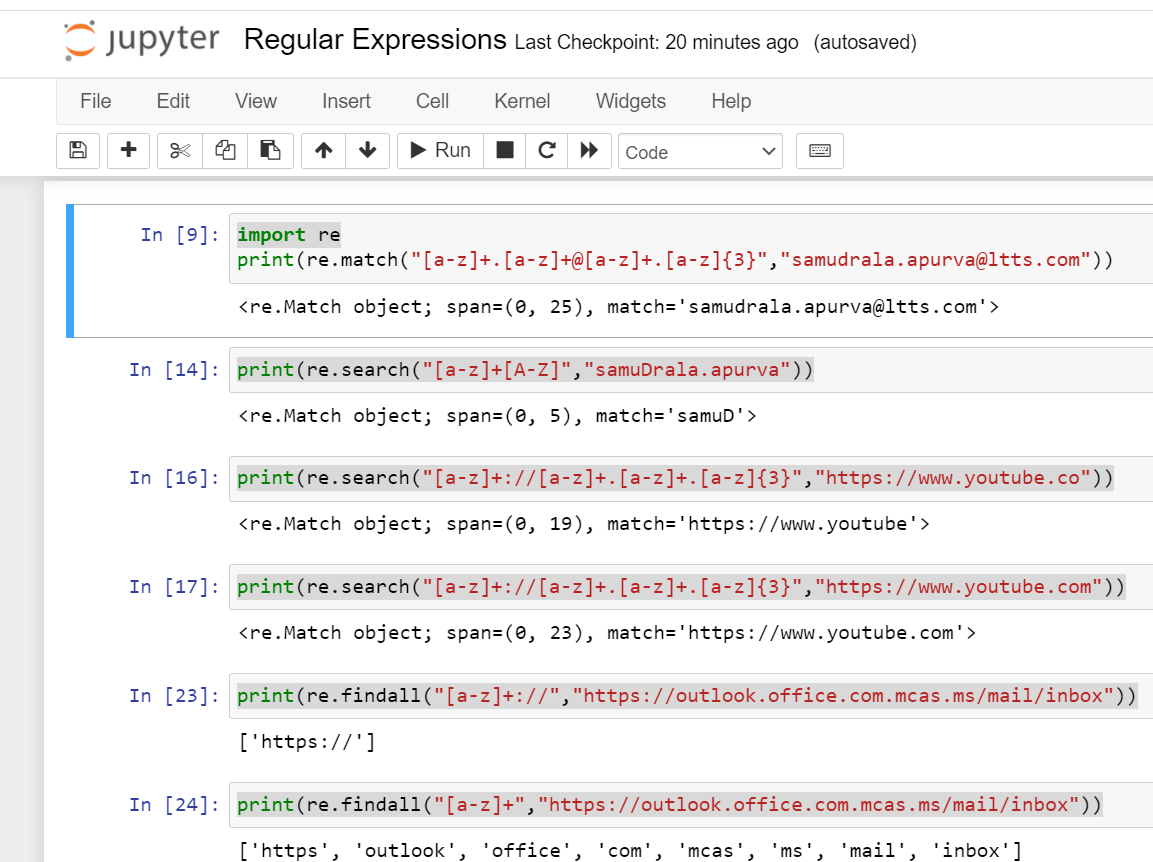


Figure 3 Regular Expressions