**Experiment No.: 57 Date:21-11-22**

**Aim:**

Write a python program to create a package (Engg), sub-package( years),modules (sem) and

create staff and student function to module?

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

**#function in Engg package file with sub as sub package file and sem as module.**

def staff(name):

   return name

def student(name2):

   return name2

**#main.py**

import Engg

import Engg.sub.sem as f

staff\_name=input("Enter the name of Staff : ")

print("The name of the staff is: ")

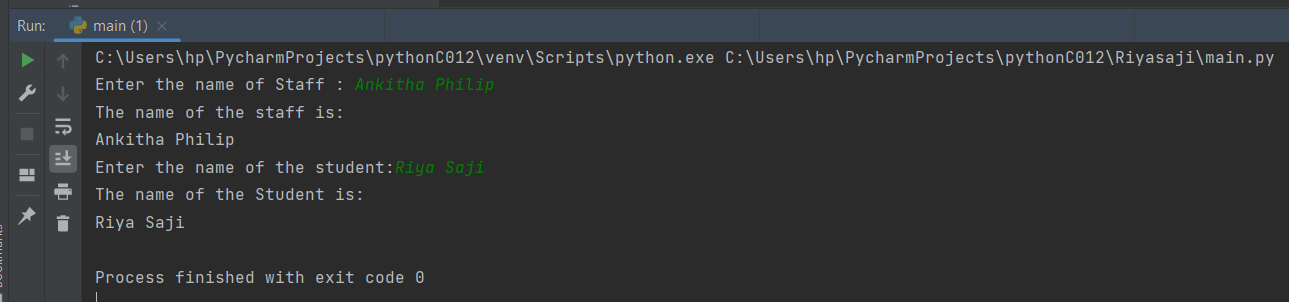
print(f.staff(staff\_name))

student\_name=input("Enter the name of the student:")

print("The name of the Student is: ")

print(f.student(student\_name))

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 58 Date:21-11-22**

**Aim:**

Write a Python program to generate a random color hex, a random alphabetical string, random value between two integers (inclusive) and a random multiple of 7 between 0 and 70.

Use random.randint()

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

import random

import string

print("Generate a random color hex:")

print("#{:06x}".format(random.randint(0, 0xFFFFFF)))

print("\nGenerate a random alphabetical string:")

max\_length = 255

s = ""

for i in range(random.randint(1, max\_length)):

    s += random.choice(string.ascii\_letters)

print(s)

print("Generate a random value between two integers, inclusive:")

print(random.randint(0, 10))

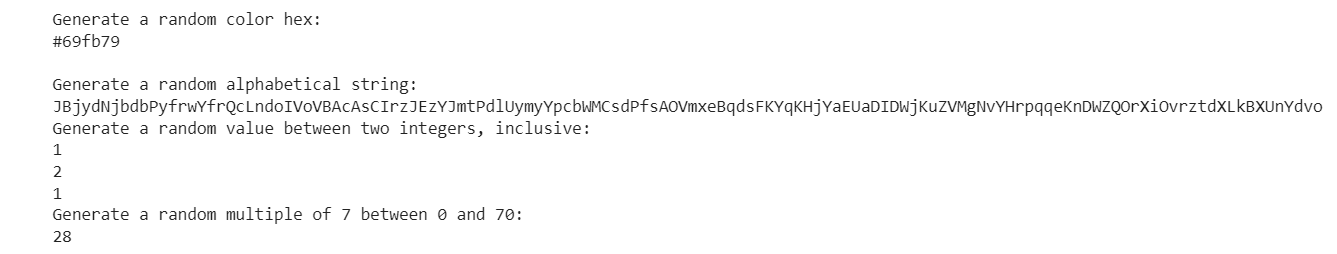
print(random.randint(-7, 7))

print(random.randint(1, 1))

print("Generate a random multiple of 7 between 0 and 70:")

print(random.randint(0, 10) \* 7)

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 59 Date:21-11-22**

**Aim:**

Write a Python program to select a random element from a list, set, dictionary (value) and a file from a directory.

Use random.choice()

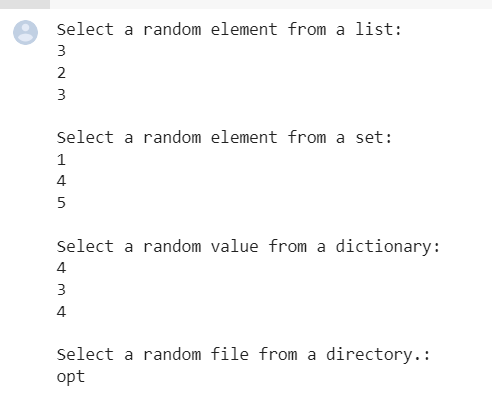
**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

import random  
import os  
print("Select a random element from a list:")  
elements = [1, 2, 3, 4, 5]  
print(random.choice(elements))  
print(random.choice(elements))  
print(random.choice(elements))  
print("\nSelect a random element from a set:")  
elements = set([1, 2, 3, 4, 5])  
# convert to tuple because sets are invalid inputs  
print(random.choice(tuple(elements)))  
print(random.choice(tuple(elements)))  
print(random.choice(tuple(elements)))  
print("\nSelect a random value from a dictionary:")  
d = {"a": 1, "b": 2, "c": 3, "d": 4, "e": 5}  
key = random.choice(list(d))  
print(d[key])  
key = random.choice(list(d))  
print(d[key])  
key = random.choice(list(d))  
print(d[key])  
print("\nSelect a random file from a directory.:")  
print(random.choice(os.listdir("/")))

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 60 Date:21-11-22**

**Aim:**

Write a Python program to create a list of random integers and randomly select multiple items from the said list. Use random.sample()

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

import random

print("Create a list of random integers:")

population = range(0, 100)

nums\_list = random.sample(population, 10)

print(nums\_list)

no\_elements = 4

print("\nRandomly select",no\_elements,"multiple items from the said list:")

result\_elements = random.sample(nums\_list, no\_elements)

print(result\_elements)

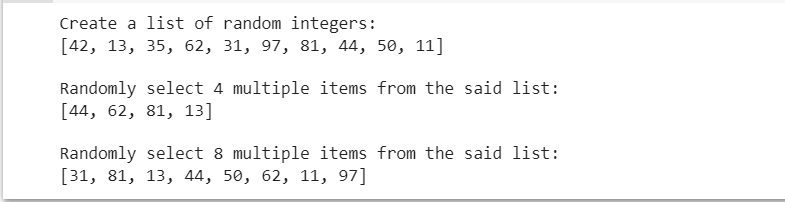
no\_elements = 8

print("\nRandomly select",no\_elements,"multiple items from the said list:")

result\_elements = random.sample(nums\_list, no\_elements)

print(result\_elements)

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 61 Date:21-11-22**

**Aim:**

**Write a Python script to display the various Date Time formats.**

a) Current date and time

b) Current year

c) Month of year

d) Week number of the year

e) Weekday of the week

f) Day of year

g) Day of the month

h) Day of week

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

import time

import datetime

print("Current date and time: " , datetime.datetime.now())

print("Current year: ", datetime.date.today().strftime("%Y"))

print("Month of year: ", datetime.date.today().strftime("%B"))

print("Week number of the year: ", datetime.date.today().strftime("%W"))

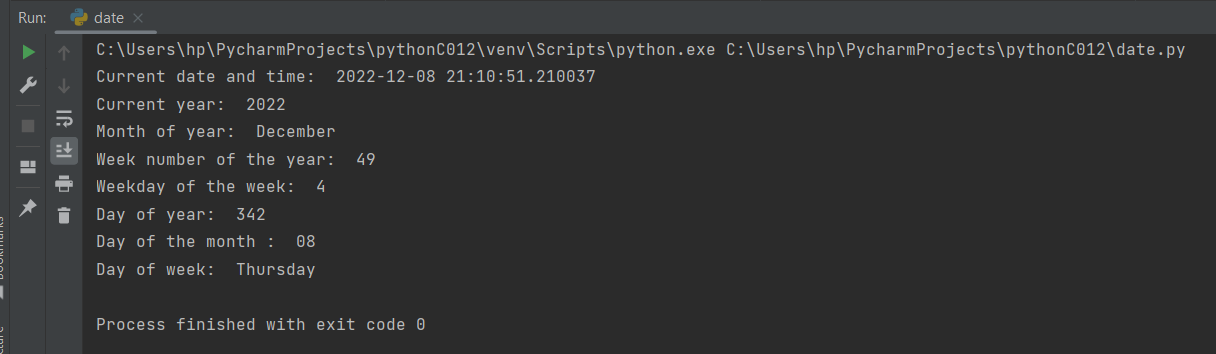
print("Weekday of the week: ", datetime.date.today().strftime("%w"))

print("Day of year: ", datetime.date.today().strftime("%j"))

print("Day of the month : ", datetime.date.today().strftime("%d"))

print("Day of week: ", datetime.date.today().strftime("%A"))

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 62 Date:21-11-22**

**Aim:**

Write a Python program to get the current time in Python.

Sample Format :  13:19:49.078205

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

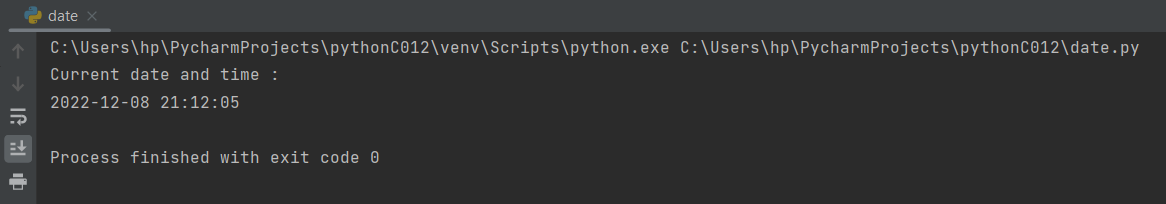
import datetime

now = datetime.datetime.now()

print ("Current date and time : ")

print (now.strftime("%Y-%m-%d %H:%M:%S"))

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 63 Date:21-11-22**

**Aim:**

Write a Python program to subtract five days from current date.

Current Date : 2015-06-22

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

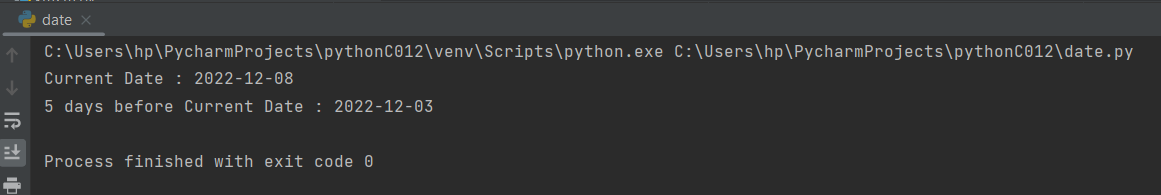
from datetime import date, timedelta

dt = date.today() - timedelta(5)

print('Current Date :',date.today())

print('5 days before Current Date :',dt)

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 64 Date:21-11-22**

**Aim:**

 Write a Python program to print next 5 days starting from today.

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

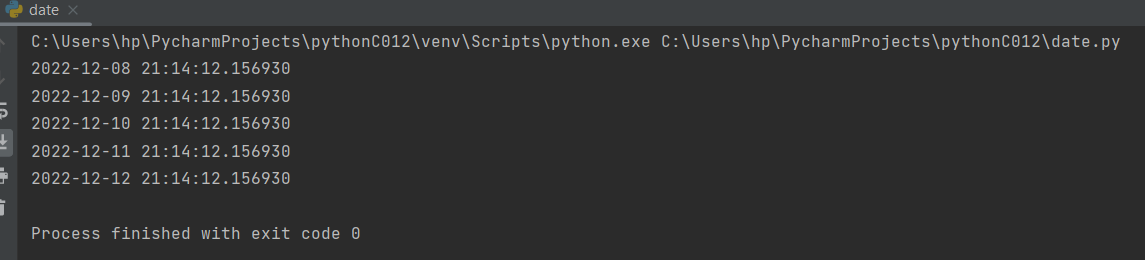
import datetime

base = datetime.datetime.today()

for x in range(0, 5):

print(base + datetime.timedelta(days=x))

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 65 Date:21-11-22**

**Aim:**

Write a Python program to add 5 seconds with the current time.

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

import datetime

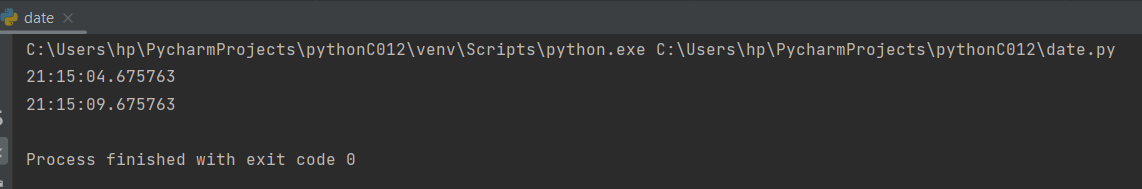
x= datetime.datetime.now()

y = x + datetime.timedelta(0,5)

print(x.time())

print(y.time())

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 65 Date:21-11-22**

**Aim:**

Write a Python program to calculate an age in year

**CO3:**

Design modules and packages - built in and user defined packages.

**Procedure:**

from datetime import date

def calculate\_age(dtob):

today = date.today()

return today.year - dtob.year - ((today.month, today.day) < (dtob.month, dtob.day))

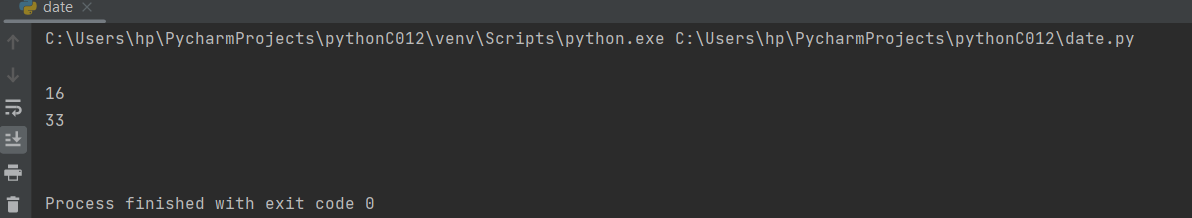
print()

print(calculate\_age(date(2006,10,12)))

print(calculate\_age(date(1989,1,12)))

print()

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 66**

**Aim:**

Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import \* statements)

**CO4:**

Implement object-oriented programming and exception handling**.**

**Procedure:**

#RECTANGLE MODULE IN GRAPHICS PACKAGE

def rectangle(l,b):  
 Rperimeter=2\*(l+b)  
 Rarea=l\*b  
 print("The perimeter of Rectangle is: ", Rperimeter, "units")  
 print("The area of Rectangle is: ",Rarea,"sq.units")

#Circle MODULE IN GRAPHICS PACKAGE

def circle(r):  
 area=3.14\*r\*r  
 perimeter=2\*3.14\*r  
 print("The Perimeter of circle is: ",perimeter," units")  
 print("The area of rectangle is: ",area,"sq.units")

#Cuboid module in Graphics package and 3d graphics sub package

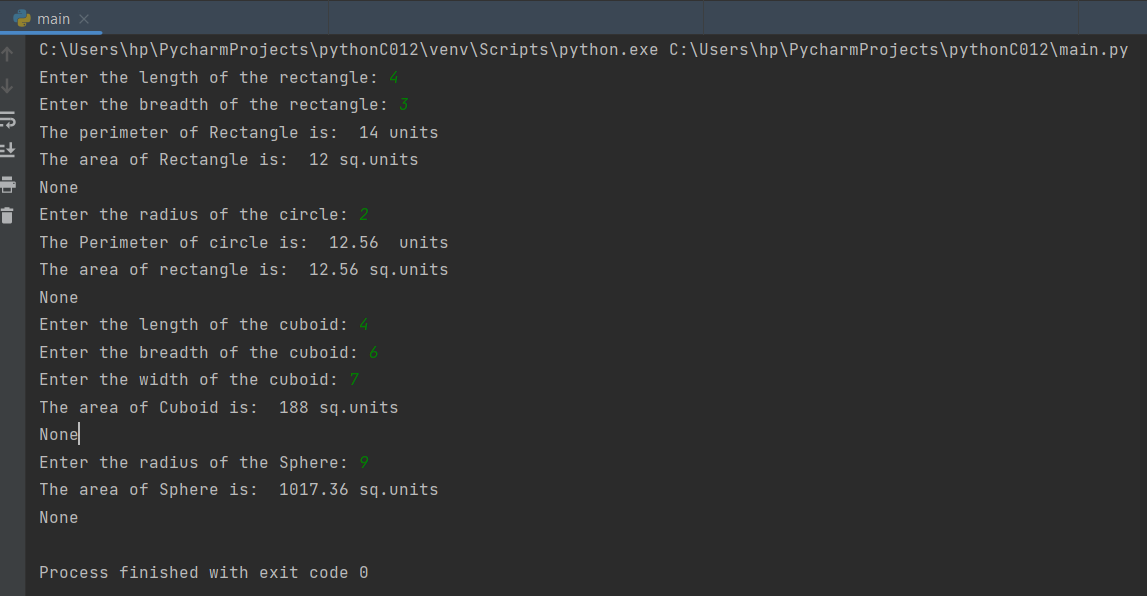
def cuboid(l,b,w):  
 Carea=2\*(l\*b+b\*w+w\*l)  
 print("The area of Cuboid is: ",Carea,"sq.units")

#Sphere module in Graphics package and 3d graphics sub packagedef sphere(r):  
 Sarea=4\*3.14\*r\*r  
 print("The area of Sphere is: ",Sarea,"sq.units")

**#main.py**

import graphics  
import graphics.rectangle as f  
l=int(input("Enter the length of the rectangle: "))  
b=int(input("Enter the breadth of the rectangle: "))  
print(f.rectangle(l,b))  
import graphics.circle as g  
r=int(input("Enter the radius of the circle: "))  
print(g.circle(r))  
  
import graphics.3Dgraphics.cuboid as h  
l1=int(input("Enter the length of the cuboid: "))  
b1=int(input("Enter the breadth of the cuboid: "))  
w1=int(input("Enter the width of the cuboid: "))  
print(h.cuboid(l1+b1+w1))  
  
import graphics.3Dgraphics.sphere as e  
r1=int(input("Enter the radius of the Sphere: "))  
print(e.sphere(r1))

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained

**Experiment No.: 67**

**Aim:**

Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

**CO4:**

Implement Object-oriented programming and exceptation handling.

**Procedure:**

class Rectangle():

   def \_\_init\_\_(self,l,w):

       self.length=l

       self.width=w

   def area(self):

       return self.length\*self.width

   def perimeter(self):

       return 2 \*(self.length+self.width)

l1=float(input("Enter the length of Rectangle1: "))

b1=float(input("Enter the breadth of Rectangle1: "))

l2=float(input("Enter the length of Rectangle2: "))

b2=float(input("Enter the breath of Rectangle1: "))

rect1=Rectangle(l1,b1)

rect2=Rectangle(l2,b2)

print("Area of Rectangle1 is {} and Perimeter of Rectangle 1 is {} :".format(rect1.area(),rect1.perimeter()))

print("Area of Rectangle2 is {} and Perimeter of Rectangle 2 is {} :".format(rect2.area(),rect2.perimeter()))

print(rect1.area()>rect2.area())

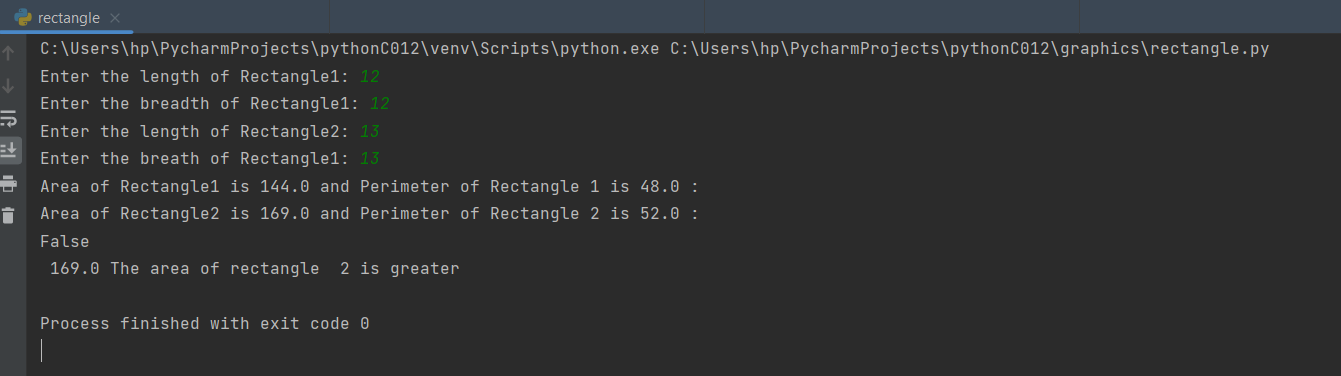
if rect1.area()>rect2.area() :

   print(" {} The area of rectangle 1 is greater".format(rect1.area()))

else:

   print(" {} The area of rectangle  2 is greater".format(rect2.area()))

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained

**Experiment No.: 68**

**Aim:**

Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank

**CO4:**

Implement Object-oriented programming and exceptation handling.

**Procedure:**

class BankAccount:

   def \_\_init\_\_(self):

       self.balance=0

       print("Hello!Welcome to the Deposit & withdrawal Machine")

   def deposit(self):

       amount=float(input("Enter Amount to be Deposit: "))

       self.balance+=amount

       print("\n Amount Deposited:",amount)

   def withdrawal(self):

       amount=float(input("Enter the Amount to be Withdrawn: "))

       if self.balance>=amount:

           self.balance-=amount

           print("\n You Withdraw:",amount)

       else:

           print("Insufficient Balance")

   def display(self):

       print("/n Net Available Balance",self.balance)

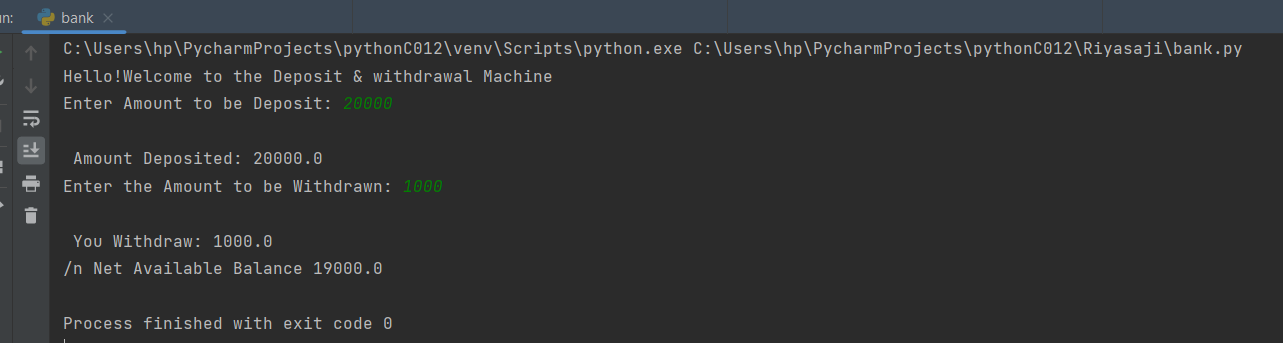
s=BankAccount()

s.deposit()

s.withdrawal()

s.display()

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained

**Experiment No.: 69**

**Aim:**

Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no\_of\_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

**CO4:**

Implement Object-oriented programming and exceptation handling.

**Procedure:**

class publisher:

   def \_\_init\_\_(self,pubname):

       self.pubname=pubname

   def display(self):

       print("Publisher name is :",self.pubname)

class book(publisher):

   def \_\_init\_\_(self,pubname,title,author):

       publisher.\_\_init\_\_(self,pubname)

       self.title=title

       self.author=author

   def display(self):

       print("Title:",self.title)

       print("Author:",self.author)

class python(book):

   def \_\_init\_\_(self,pubname,title,author,price,no\_of\_pages):

       book.\_\_init\_\_(self,pubname,title,author)

       self.price=price

       self.no\_of\_pages=no\_of\_pages

   def display(self):

       print("Title : ",self.title)

       print("Author : ",self.author)

       print("Publisher : ",self.pubname)

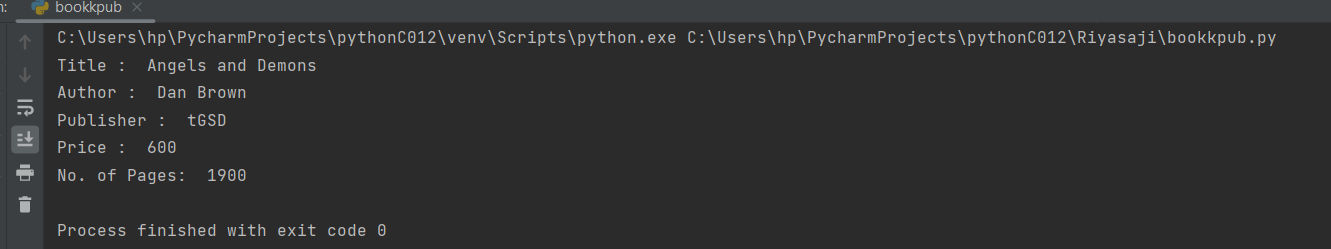
       print("Price : ", self.price)

       print("No. of Pages: ",self.no\_of\_pages)

b1=python("tGSD","Angels and Demons","Dan Brown",600,1900)

b1.display()

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained

**Experiment No.: 70**

**Aim:**

Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find a sum of 2 times.

**CO4:**

Implement Object-oriented programming and exceptation handling.

**Procedure:**

class Time:

def \_init\_(self, hour, minute, second):

self.\_\_hour = hour

self.\_\_minute = minute

self.\_\_second = second

def \_add\_(self, other):

t3.\_hour = t1.hour + t2.\_hour

t3.\_minute = t1.minute + t2.\_minute

t3.\_second = t1.second + t2.\_second

if t3.\_\_second > 59:

t3.\_\_second -= 60

t3.\_minute = t3.\_minute + 1

if t3.\_\_minute > 59:

t3.\_\_minute -= 60

t3.\_hour = t3.\_hour + 1

return str(t3.\_hour) + ':' + str(t3.minute) + ':' + str(t3.\_second)

h1 = int(input("Enter the hour"))

m1 = int(input("Enter the minute"))

s1 = int(input("Enter the second"))

h2 = int(input("Enter the hour"))

m2 = int(input("Enter the minute"))

s2 = int(input("Enter the second"))

t1 = Time(h1, m1, s1)

t2 = Time(h2, m2, s2)

t3 = Time(0, 0, 0)

print(t1 + t2)

**Output Screenshot:**

**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained

**Experiment No.: 70**

**Aim:**

Create a class Time with private attributes hour, minute and second. Overload ,compare the two times.

**CO4:**

Implement Object-oriented programming and exceptation handling.

**Procedure:**

class Time:  
 def \_init\_(self, hour, minute, second):  
 self.\_\_hour = hour  
 self.\_\_minute = minute  
 self.\_\_second = second  
  
 def \_add\_(self, other):  
 return 'Time is: ' + str(self.\_hour + other.hour) + ':' + str(self.minute + other.\_minute) + ':' + str(  
 self.\_second + other.\_second)  
  
 def \_lt\_(self, other):  
 if self.\_hour < other.\_hour:  
 return "true"  
 elif self.\_hour == other.\_hour:  
 if self.\_minute < other.\_minute:  
 return "true"  
 elif self.\_minute == other.\_minute:  
 if self.\_second < other.\_second:  
 return "true"  
 else:  
 return "false"  
 else:  
 return "false"  
  
  
h1 = int(input("Enter the hour"))  
m1 = int(input("Enter the minute"))  
s1 = int(input("Enter the second"))  
h2 = int(input("Enter the hour"))  
m2 = int(input("Enter the minute"))  
s2 = int(input("Enter the second"))  
t1 = Time(h1, m1, s1)  
t2 = Time(h2, m2, s2)  
print(t1 + t2)  
print(t1 < t2)

**Output Screenshot:**

**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained

**Experiment No.: 71**

**Aim:**

Create a class Time with private attributes hour, minute and second. Overload ,compare the two times.

**CO4:**

Implement Object-oriented programming and exceptation handling.

**Procedure:**

class Time:  
    def \_init\_(self, hour, minute, second):  
        self.\_\_hour = hour  
        self.\_\_minute = minute  
        self.\_\_second = second  
  
    def \_add\_(self, other):  
        return 'Time is: ' + str(self.\_hour + other.hour) + ':' + str(self.minute + other.\_minute) + ':' + str(  
            self.\_second + other.\_second)  
  
    def \_lt\_(self, other):  
        if self.\_hour < other.\_hour:  
            return "true"  
        elif self.\_hour == other.\_hour:  
            if self.\_minute < other.\_minute:  
                return "true"  
            elif self.\_minute == other.\_minute:  
                if self.\_second < other.\_second:  
                    return "true"  
                else:  
                    return "false"  
        else:  
            return "false"  
  
  
h1 = int(input("Enter the hour"))  
m1 = int(input("Enter the minute"))  
s1 = int(input("Enter the second"))  
h2 = int(input("Enter the hour"))  
m2 = int(input("Enter the minute"))  
s2 = int(input("Enter the second"))  
t1 = Time(h1, m1, s1)  
t2 = Time(h2, m2, s2)  
print(t1 + t2)  
print(t1 < t2)

**Output Screenshot:**

**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 72**

**Aim:**

Write a Python program to read a file line by line and store it into a list.

**CO5:**

Create files and form regular expressions for effective search operations on strings and files..

**Procedure:**

#textfile

Amal Jyothi College of Engineering

Kovapally P O

Kottayam - 686510

Kerala

#python file

# Program to read text file and store it in list

# using readlines() method

openFiles = open('riya.txt','r')

fileLines=openFiles.readlines()

print("\n Content Of Files(using readlines() method): \n")

print(fileLines)

# by using strip method

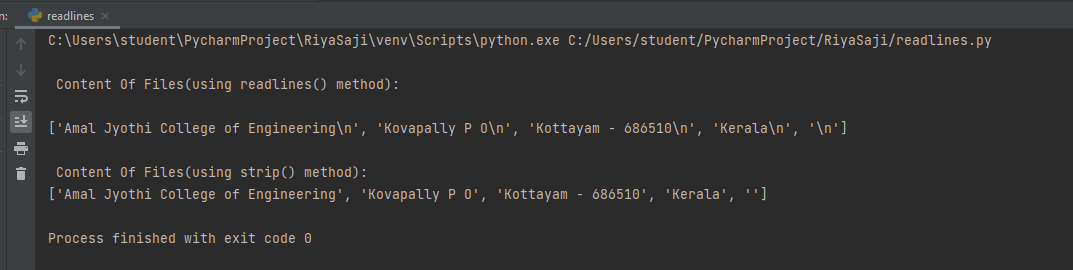
fileLines=[x.strip() for x in fileLines]

print("\n Content Of Files(using strip() method): ")

print([x.strip() for x in fileLines])

openFiles.close()  # closing the file

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 73**

**Aim:**

Python program to copy odd lines of one file to other

**CO5:**

Create files and form regular expressions for effective search operations on strings and files..

**Procedure:**

#riya.txt

Amal Jyothi College of Engineering

Kovapally P O

Kottayam - 686510

Kerala

#python file

# Program to copy odd lines of one file to another

# Opening files for reading and writing data

input\_file=open('riya.txt')

output\_file=open('WriteData.txt','w')

# Copying /reading contents from read\_file to copy\_data

copy\_data=input\_file.readlines()

print("\n Actual file is : ")

print(copy\_data,"\n")

for i in range(0,len(copy\_data)):

if i%2==0:

output\_file.write(copy\_data[i])

else:

pass

# Closing file after writing

output\_file.close()

# Opening write file in read mode and printing values

output\_file=open('WriteData.txt','r')

print("\n Odd Lines Are : ")

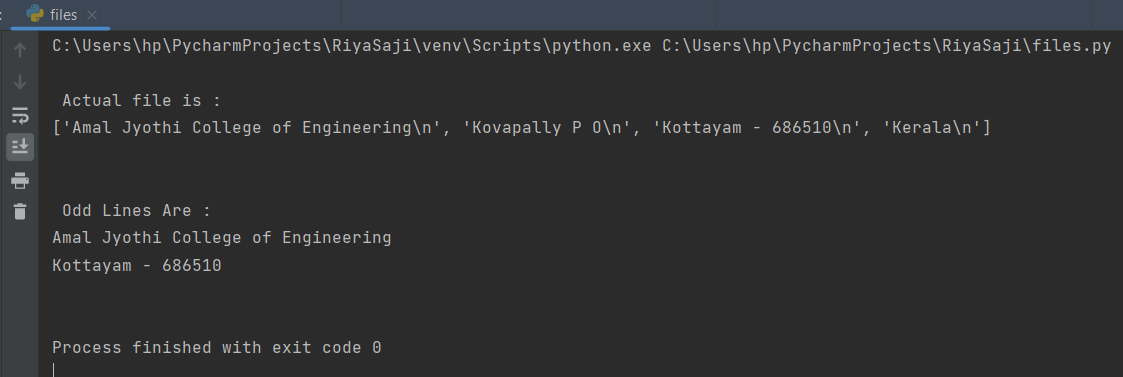
print(output\_file.read())

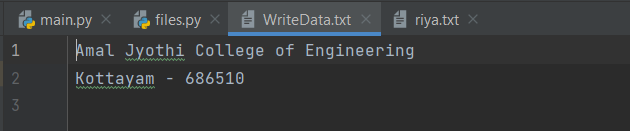
# Closing files

input\_file.close()

output\_file.close()

**Output Screenshot:**





**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 74**

**Aim:**

Write a Python program to read each row from a given csv file and print a list of strings.

**CO5:**

Create files and form regular expressions for effective search operations on strings and files..

**Procedure:**

#python file

import csv

# Open the CSV file

with open('excelsheet.csv','r') as file:

# Create a CSV reader

reader=csv.reader(file)

# Iterate over the rows of the CSV file

for row in reader:

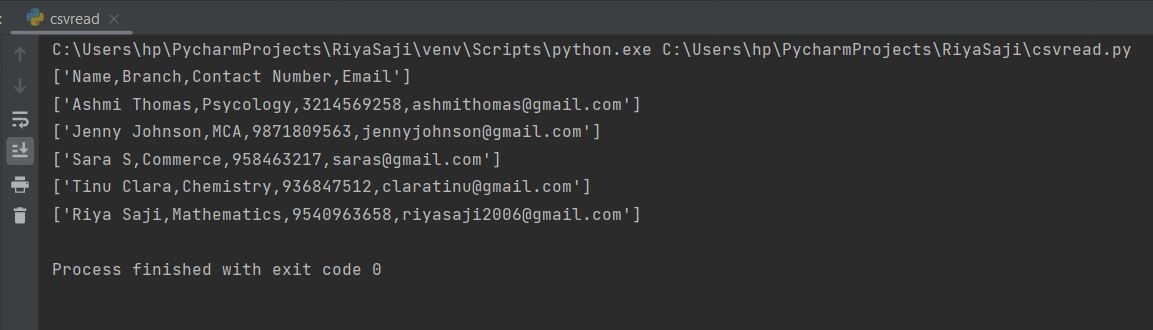
# Print the rows as a list of strings

print(row)

#excelsheet.csv

"Name,Branch,Contact Number,Email"  
"Ashmi Thomas,Psycology,3214569258,ashmithomas@gmail.com"  
"Jenny Johnson,MCA,9871809563,jennyjohnson@gmail.com"  
"Sara S,Commerce,958463217,saras@gmail.com"  
"Tinu Clara,Chemistry,936847512,claratinu@gmail.com"  
"Riya Saji,Mathematics,9540963658,riyasaji2006@gmail.com"

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 75**

**Aim:**

Write a Python program to read specific columns of a given CSV file and print the content of the columns.

**CO5:**

Create files and form regular expressions for effective search operations on strings and files..

**Procedure:**

import csv

# Specify the column indices that you want to raed

# e.g column 0 is the first column 1 is the second column , etc. .

columns\_to\_read=[0,2]

#open the csv file and read the contents

with open('data.csv','r') as f:

clmn\_reader=csv.reader(f)

# Iterate over the rows of the CSV file

for row in clmn\_reader:

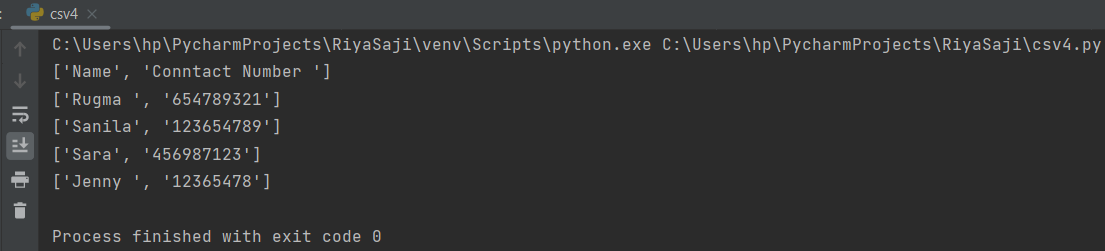
# Print the contents of the specified columns

print([row[i] for i in columns\_to\_read])

**#csvfile**

Name,Branch ,Conntact Number ,email  
Rugma ,B.Com,654789321,rugma123@gmail.com  
Sanila,BCA,123654789,sanilasunny@gmail.com  
Sara,B.com,456987123,ssara@gmail.com  
Jenny ,BCA,12365478,johsonjenny@gmail.com

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO5 was attained.

**Experiment No.: 76**

**Aim:**

Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

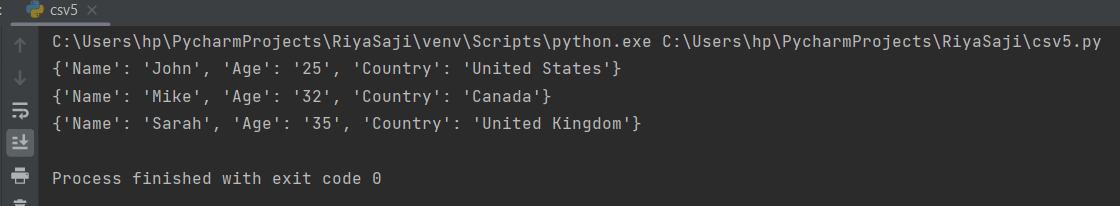
**CO5:**

Create files and form regular expressions for effective search operations on strings and files..

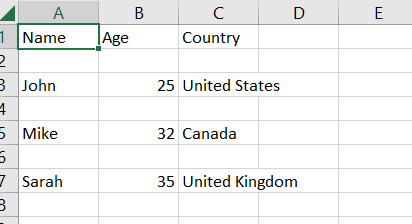
**Procedure:**

# Write a Python program to write a Python dictionary to a csv file.  
# After writing the CSV file read the CSV file and display the content.  
  
import csv  
  
# Data to be inserted  
data = [{'Name': 'John', 'Age': 25, 'Country': 'United States'},  
{'Name': 'Mike', 'Age': 32, 'Country': 'Canada'},  
{'Name': 'Sarah', 'Age': 35, 'Country': 'United Kingdom'}]  
  
# Write to CSV file  
with open('people.csv', 'w') as csvfile:  
 headernames = ['Name', 'Age', 'Country']  
 csvwriter = csv.DictWriter(csvfile, fieldnames=headernames)  
 csvwriter.writeheader()  
 for row in data:  
 csvwriter.writerow(row)  
  
# Read from CSV file and print contents  
with open('people.csv', 'r') as csvfile:  
 reader = csv.DictReader(csvfile)  
 for row in reader:  
 print(row)

**Output Screenshot:**



**#people.csv**



**Result**

The program was executed and the result was successfully obtained. Thus CO5 was attained.