**Lingaya’s Vidyapeeth, Faridabad**

(Deemed to be University under Section 3 of UGC Act, 1956)

**Python Programming Lab**

**CS-255**

**LAB File**

**B.TECH 2nd Year [C.S.E]**



Nachauli, Old Faridabad-Jasana Road,

Faridabad 121002, Phone: (129) 2201008, 2201009;

WebSite: www.lingayasuniversity.edu.in

Submitted to:SubmittedBy**:**

Mr. Monu NAME –S Kashaf Alam

ROLL No – 22CS75CL

SEMESTER – 3rd

**VISION:**

To bring forth cultured graduates meeting the expectation of national and multi-national industries exceling in the field of computing as well as in higher studies and research.

**MISSION:**

1. To provide strong theoretical knowledge of computer science with practical training which meets the industries expectations.
2. To train necessary skills to further higher studies and professional growth.
3. To inculcate ethical valued in graduates through various social-cultural activities.

**Course Objectives:**

To build programming logic and thereby developing skills in problem solving using Python programming language; To be able to do testing and debugging of code written in Python Emphasize the concepts and constructs rather than on language features.

**Course Outcomes (COs):**

After completion of course, students would be able to:

* 1. CO1: To learn and understand Python programming basics and paradigm.
  2. CO2: Define and demonstrate the use of built-in data structures “lists” and “dictionary”
  3. CO3: Design and implement a program to solve a real world problem
  4. CO4: Solve exception handling problem and files
  5. CO5: Make database connectivity in python programming language

|  |  |
| --- | --- |
| **Sl.no** | **List of Experiments** |
| **1** | **Program using if else structure** |
| a) | Find the largest among three numbers |
| b) | Python program to check leap year |
| c) | Python program to take in the marks of 5 subjects and display the grade |
|  |  |
| **2** | **Program using for and while loop** |
| a) | Python program to check whether given number is prime or not |
| b) | Python program to find LCM of two number |
| c) | Write a python program to compute the GCD of two numbers |
| d) | Python program to find the sum of digits in a number |
| e) | Python program to convert binary number to decimal number |
| f) | Python program to display Fibonacci series |
| **3** | **Program using list and string data structure** |

|  |  |
| --- | --- |
| a) | Write Python Program to input a list of integers, (1) display the no of elements in the list  (2) display minimum and maximum element in the list (3) display sum of square of all the element in the list (4) add a new element at end and display the list (5) add a new element at given index and display list (6) display the occurrence of given element in the list (7) remove the given element in the list (8) add element from a new list from given list (9) sort the given list & reverse the given list (10) also perform slicing, concatenation and multiplication operation b |
| b) | A fruit seller sells different type of fruits. Type of fruits and corresponding rates are stored in two different lists. Customer can order any type of fruit (one or more type) in any quantity. If total bill of customer is greater than 500, customer is given 10% discount. If any of the fruits required by the customer is not available in the store, then consider the bill  amount to be -1. Write a Python program to calculate and display the bill amount. |
| 4) | **Programs using concept of sets, tuple and dictionary** |
| a) | Write a Python program that take a string as input and store the character and occurrence of each character in a dictionary. Create two lists from dictionary first having each character  in sorted order of their frequency and second having corresponding frequency. |
| b) | A furniture seller sells different type of furniture, Type of Furniture and rates are stored in a dictionary. Customer can order any type of furniture (one or more type) in any quantity. If total bill of customer is greater than 10,000, customer is given 5% discount. 8% GST is charged on total bill. If any of the furniture required by the customer is not available in the  store, then consider the bill amount to be -1. Write a Python program to calculate and display the bill amount. |
| c) | Consider a scenario from Lingayas Vidyapeeth. Given below are two Sets representing the names of students enrolled for a particular course: java\_course = {"Anmol", "Rahul", "Priyanka", "Pratik"} python\_course = {"Rahul", "Ram", "Nazim", "Vishal"}Write a Python program to list the number of students enrolled for:1)Python course2)Java course only3)Python course only4)Both Java and Python courses5)Either Java or Python courses  but not both6)Either Java or Python. |
| 5 | **Using function in python** |
| a) | Write Python functions using the concept of Keyword & default arguments and write a  program to use them. |
| b) | Write python functions to use the concept of variable length argument & global variable.  Write a program to use these functions. |
| 6 | **Program using concept of class object, class variable,class method,static method** |
| a) | Create a class Account with name, account no and balance as attribute and no\_of\_accounts as class variable. Account no should be generated automatically (starting from 1) using the class variable no\_of\_account. Add the methods for displaying the account information, depositing given amount, withdrawing given amount and initializer method to initialize the object. Create objects of Account class and call different method to test the class. |
| b) | Create a class Employee with name , empid ,salary as attribute and no\_of\_ employee and annual\_incr ( % annual increment) as class variable. empid should be generated automatically (starting from 1) using the class variable, no\_of\_employee. Add the instance methods for displaying the employee information, annually increasing the salary with help of class variable annual\_incr , class method to change the value of annual\_incr and  initializer method to initialize the object. Create objects of employee class and call different method to test the class (program using class method) |

|  |  |
| --- | --- |
| c) | Write a Program to showing the use of built in class attributes ( doc , dict ,  name , module , bases ) and special methods( del ( ), str ( )) and built in function is instance(). |
| 7 | **Program using the concept of Inheritance** |
| a) | Create a class Polygon to represent a polygon having no of sides and a list having magnitude of each side as attribute. Add the inputSides() to input sides and displaySides() to display sides as methods. Derive a class Triangle from Polygon and add an additional method displayArea() to display area. Create object of Triangle and call different methods to test  the class. |
| b) | Create a class Person having name, age, as attributes, init () method to initialize the object and display() to display person information. Derive a class Student from Person having roll no, University name, branch as additional attributes and init (), display() to display student information and change\_Branch() method. Create object of Student type and call different methods to test the class. |
| c) | Write a program to show the concept of multiple inheritance in python. |
| 8 | **Program using the concept of Polymorphism, operator overloading** |
| a) | In a retail outlet there are two modes of bill Payment (1) Cash : Calculation includes VAT(10%) Total Amount = Purchase amount + VAT (2) Credit card: Calculation includes processing charge and VAT Total Amount = Purchase amount + VAT (10%) + Processing charge (2%) The act of bill payment is  same but the formula used for calculation of total amount differs as per the mode of payment. Can the Payment maker simply call a method and that method dynamically selects the formula for the total amount? Demonstrate this Polymorphic behaviour with code. |
| b) | Write a program to create a class to represent length in feet and inch. Overload the “+”  operator to add the two object of length type. |
| c) | Write a program to overload comparison operator in python. |
| 9 | **Program on file handling in python** |
| a) | Write a python program to write few lines on a file, read it back and create a dictionary having each word in file as keys in dictionary and occurrence of these word as values and  print the dictionary. |
| b) | A file student.txt store student information. Information about each student is written on separate line in the form: roll-no student-name (student-name may consist of any number of words).Write a Python program that takes student roll no as input and print the student  name. If roll no is not present in the file it display: “roll no not present in the file”. |
| c) | Write a python program to read a file that contains email ids on the separate lines in the form: “[personname@companyname.com.](mailto:personname@companyname.com) Create a new file that contain only company names, read the new file to print the company name. |
| 10 | **Program on Exception handling** |
| a) | Write a function divide (arg1, arg2) to divide arg1 by arg2. Use the exception handling mechanism to handle all type of possible exceptions that may occur. Take the value of arg1 and arg2 (of any type) from user as input and call the function divide to print the result of division or suitable message if any type of exception occurs (use also else and finally block). |

|  |  |
| --- | --- |
| b) | Write a program to open a file in read only mode read data from file and then try to write data on file. Use the exception handling mechanism to handle all type of possible exception. |

**Hardware Requirements:**

* Modern Operating System:
  + Windows 7 or 10
  + Mac OS X 10.11 or higher, 64-bit
  + Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
* x86 64-bit CPU (Intel / AMD architecture)
* 4 GB RAM
* 5 GB free disk space

**Software Requirements:**

* Python 3.9 or any version of Python 3

# PROGRAM – 1

**AIM :- Programs using if else structure**

1. Find the Largest Among Three Numbers
2. Python Program to Check Leap Year
3. Python Program to Take the Marks of 5 Subjects and Display the Grade
4. Python Program to Check if a Date is Valid and Print next date
5. **Find the Largest Among Three Numbers**

**Code:-**

num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: ")) num3 = float(input("Enter third number: "))

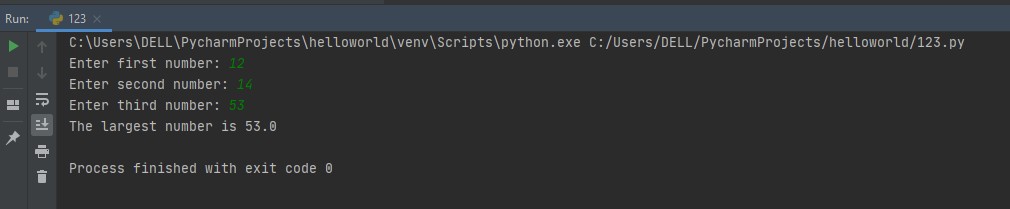
if (num1 > num2) and (num1 > num3): largest = num1

elif (num2 > num1) and (num2 > num3): largest = num2

else:

largest = num3

print("The largest number is", largest)

**Output:-**

1. **Python Program to Check Leap Year**

**Code:-**

year = int(input("Enter the year: ")) if year%4==0:

if year%100==0:

if year % 400 == 0:

print("{0} is a leap year!".format(year)) else:

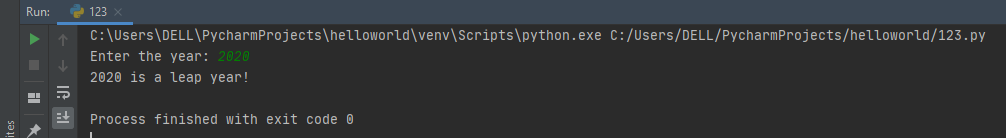
print("{0} is not a leap year!".format(year))

else:

print("{0} is a leap year!".format(year))

else:

print("{0} is not a leap year!".format(year))

**Output:-**

1. **Python Program to Take the Marks of 5 Subjects and Display the Grade Code:-**

sub1=int(input("Enter marks of the first subject: ")) sub2=int(input("Enter marks of the second subject: ")) sub3=int(input("Enter marks of the third subject: ")) sub4=int(input("Enter marks of the fourth subject: ")) sub5=int(input("Enter marks of the fifth subject: "))

avg=(sub1+sub2+sub3+sub4+sub4)/5 if(avg>=90):

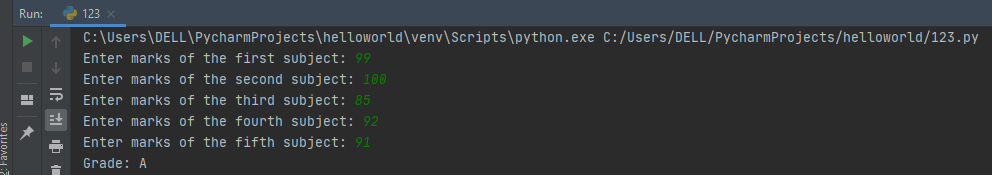
print("Grade: A") elif(avg>=80 and avg<90):

print("Grade: B") elif(avg>=70 and avg<80):

print("Grade: C") elif(avg>=60&avg<70):

print("Grade: D") else:

print("Grade: F")

**Output:-**

1. **Python Program to Check if a Date is Valid and Print next date Code:-**

date=input("Enter the date: ") dd,mm,yy=date.split('-') dd=int(dd)

mm=int(mm) yy=int(yy)

if(mm==1 or mm==3 or mm==5 or mm==7 or mm==8 or mm==10 or mm==12): max1=31

elif(mm==4 or mm==6 or mm==9 or mm==11): max1=30

elif(yy%4==0 and yy%100!=0 or yy%400==0): max1=29

else:

max1=28

if(mm<1 or mm>12): print("Date is invalid.")

elif(dd<1 or dd>max1): print("Date is invalid.")

elif(dd==max1 and mm!=12): dd=1

mm=mm+1

print("The incremented date is: ",dd,mm,yy) elif(dd==31 and mm==12):

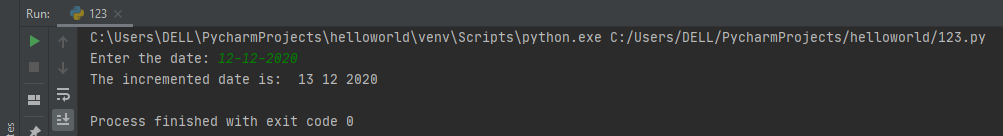
dd=1 mm=1 yy=yy+1

print("The incremented date is: ",dd,mm,yy) else:

dd=dd+1

print("The incremented date is: ",dd,mm,yy)

**Output:-**



**Program 2**

**AIM :- Programs using for and while loop**

* 1. Python Program to check whether given number is Prime Number or not
  2. Python Program to Find LCM of two numbers
  3. Write a Python program to compute the GCD of two numbers
  4. Python Program to Find the Sum of Digits in a Number
  5. Python Program to convert binary number to decimal number
  6. Python Program to Display Fibonacci sequence Using Recursion

1. **Python Program to check whether given number is Prime Number or not**

**Code:-**

n = int(input("ENTER THE NUMBER:"))

if n>1:

for i in range(2, n): if n%i ==0:

print(n,"is not a prime number!")

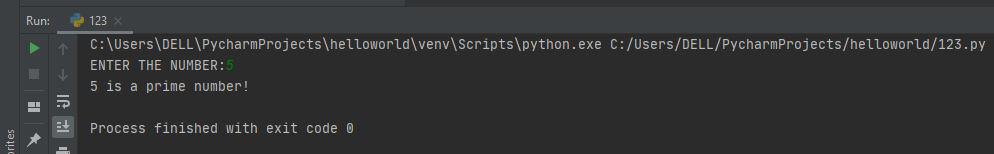
break

else:

print(n,"is a prime number!")

else:

print("It is not a a prime number!")

**Output:-**

1. **Python Program to Find LCM of two numbers**

**Code:-**

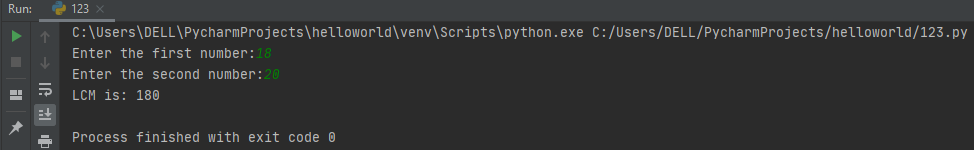
a=int(input("Enter the first number:")) b=int(input("Enter the second number:")) if(a>b):

min1=a else:

min1=b while(1):

if(min1%a==0 and min1%b==0): print("LCM is:",min1)

break min1=min1+1

**Output:-**

1. **Write a Python program to compute the GCD of two numbers code:-**

def compute(x,y): if x>y:

small=y else:

small=x

for i in range(1,small+1):

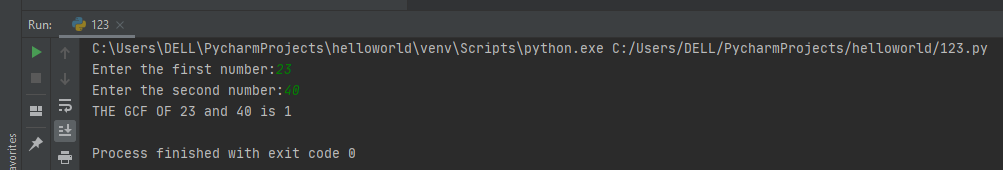
if((x%i==0)and(y%i==0)): gcf=i

return gcf

num1=int(input("Enter the first number:")) num2=int(input("Enter the second number:"))

print("THE GCF OF",num1,"and",num2,"is",compute(num1,num2))

**Output:-**



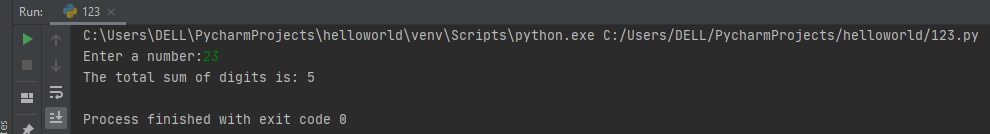
1. **Python Program to Find the Sum of Digits in a Number Code:-**

n=int(input("Enter a number:")) tot=0

while(n>0): dig=n%10 tot=tot+dig n=n//10

print("The total sum of digits is:",tot)

**Output:-**



1. **Python Program to convert binary number to decimal number Code :-**

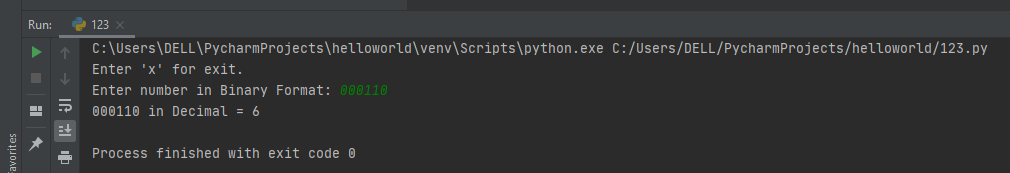
print("Enter 'x' for exit.");

binary = input("Enter number in Binary Format: "); if binary == 'x':

exit(); else:

decimal = int(binary, 2); print(binary,"in Decimal =",decimal);

**Output:-**



1. **Python Program to Display Fibonacci sequence Using Recursion Code:-**

def recur\_fibo(n): """Recursive function to print Fibonacci sequence""" if n <= 1:

return n else:

return(recur\_fibo(n-1) + recur\_fibo(n-2)) nterms = 10

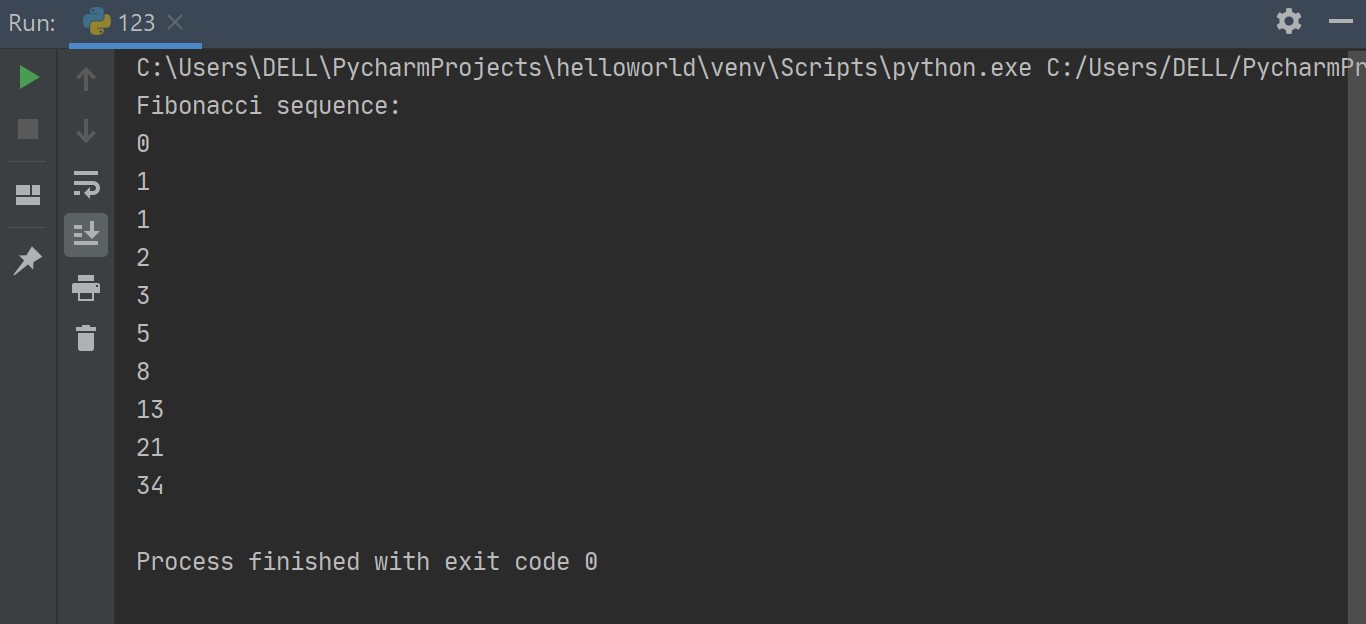
if nterms <= 0:

print("Plese enter a positive integer") else:

print("Fibonacci sequence:") for i in range(nterms):

print(recur\_fibo(i))

**Output:-**



# PROGRAM- 3

**AIM:- Program using List and String data structure**

1. Write Python Program to input a list of integers, (1) display the no of elements in the list (2) display minimum and maximum element in the list (3) display sum of square of all the element in the list (4) (5) add a new element at end and display the list (6) add a new element at given index and display list (7) display the occurrence of given element in the list (8) remove the given element in the list (9) add element from a new list from given list (10) sort the given list & reverse the given list (11) also perform slicing, concatenation and multiplication operation

**Code:-**

vowels = ['a', 'e', 'i', 'o', 'i', 'u'] count = vowels.count('i') print('The count of i is:', count) lst = [10, 20, 30, 40]

print("Maximum element in the list is :", max(lst), "\nMinimum element in the list is :", min(lst))

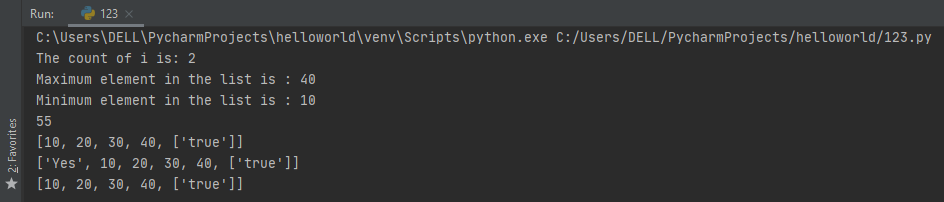
l = [1,2,3,4,5]

print(sum(i\*i for i in l))

lst.append(["true"]) print(lst) lst.insert(0,"Yes") print(lst)

lst[1:4] lst.remove("Yes") print(lst)

**Output:-**



1. **A fruit seller sells different type of fruits. Type of fruits and corresponding rates are stored in two different lists. Customer can order any type of fruit (one or more type) in any quantity. If total bill of customer is greater than 500, customer is given 10% discount. If any of the fruits required by the customer is not available in the store, then consider the bill amount to be -Write a Python program to calculate and display the bill amount.**

**Code:-** fruit\_list=['Apple','Mango','Guava','Lemon'] price\_list=[120,80,70,50]

def billcal(fruit\_list,price\_list): shopped\_items=[] quantity=[]

num=int(input("How many fruits you want to purchase?:")) bill=0

for i in range(0,num): x=shopped\_items.append(input("Enter the fruit item:")) y=quantity.append(int(input("Enter the quantity:")))

for x,y in zip(shopped\_items,quantity): if x not in fruit\_list:

return -1 else:

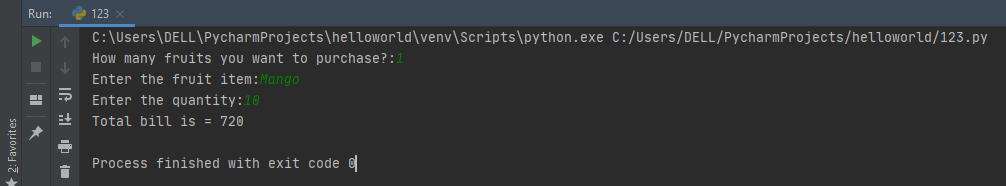
m=fruit\_list.index(x) bill+=price\_list[m]\*y

if bill>500:

bill-=(bill//10) return bill

total=billcal(fruit\_list,price\_list) print("Total bill is =",total)

**Output:-**



# PROGRAM-4

**AIM- Programs using concept of sets, tuple & dictionary**

1. Write a Python program that take a string as input and store the character and occurrence of each character in a dictionary. Create two lists from dictionary first having each character in sorted order of their frequency and second having corresponding frequency.

**Code:-** string="abrakadabraabaajabba" dict1={}

list1=[] list2=[]

for item in string: if item in dict1:

dict1[item]+=1;

else:

dict1[item]=1; print(str(dict1))

for keys in sorted(dict1, key=dict1.get, reverse=True): print(keys,dict1[keys])

list1=list(dict1.values()) list1.sort(reverse=True) for i in list1:

for j in dict1:

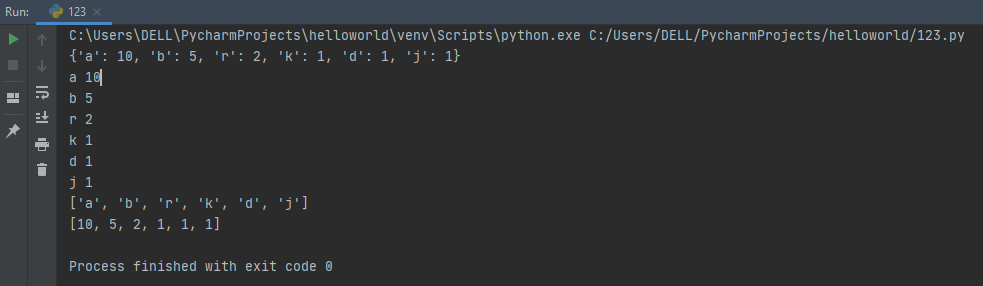
if dict1[j]==i: if j in list2: continue

else:

list2.append(j) print(list2)

print(list1)

**Output:-**



1. **A furniture seller sells different type of furniture, Type of Furniture and rates are stored in a dictionary. Customer can order any type of furniture (one or more type) in any quantity. If total bill of customer is greater than 10,000, customer is given 5% discount. 8% GST is charged on total bill. If any of the furniture required by the customer is not available in the store, then consider the bill amount to be -1. Write a Python program to calculate and display the bill amount**

**Code:-** dict1={"chair":1000,"table":7000,"desk":1500} def mart(dict1):

shopping\_list={}

n = int(input("How many items to buy: ")) bill = 0

for x in range (0,n) :

x = input("Enter the furniture item: ") shopping\_list[x] = int(input("Qunatity: "))

for item in (shopping\_list): if item not in dict1:

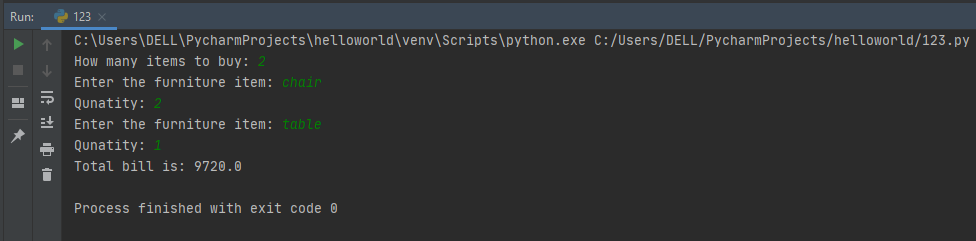
return -1 else :

bill += int(dict1[item])\*int(shopping\_list[item]) if bill > 10000 :

bill -= (bill//20) bill += (bill\*2)/25 return bill

print("Total bill is:" ,mart(dict1))

**Output:-**



1. **Consider a scenario from Lingayas Vidyapeeth. Given below are two Sets representing the names of students enrolled for a particular course: java\_course = {"Anmol", "Rahul", "Priyanka", "Pratik"} python\_course =**

**{"Rahul", "Ram", "Nazim", "Vishal"}. Write a Python program to list the number of students enrolled for:**

1. **Python course**
2. **Java course only**
3. **Python course only**
4. **Both Java and Python courses**
5. **Either Java or Python courses but not both**
6. **Either Java or Python**

**Code:-**

java\_course = {"anmol", "rahul", "priyanka","prateek"}; python\_course = {"ram", "rahul", "nazim","prateek"};

# python course

print("python course :", java\_course | python\_course)

# java course only

print("java course only :", java\_course - python\_course)

# python course only

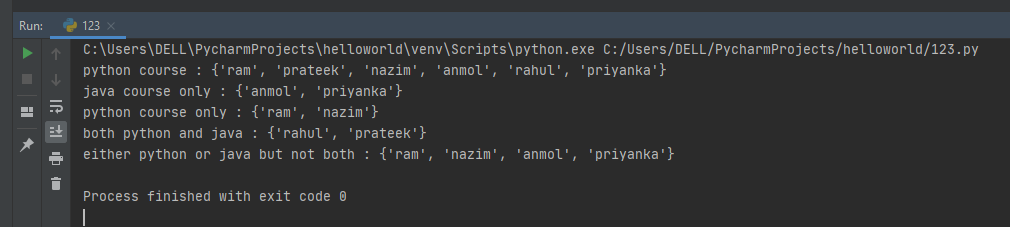
print("python course only :", python\_course - java\_course)

# both python and java

print("both python and java :", java\_course & python\_course)

# either python or java but not both

print("either python or java but not both :",java\_course ^ python\_course)

**Output**

# PROGRAM-5

**AIM- Using Function in Python**:

1. **Write Python functions using the concept of Keyword & default arguments and write a program to use them**

**Code:-**

def improper\_return\_function(a): if (a % 2) == 0:

return True else:

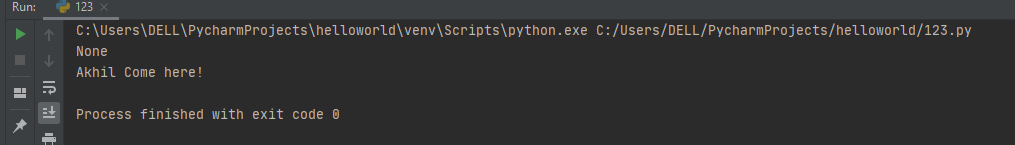
return False

x = improper\_return\_function(3) print(x)

def defaultArg(name, foo='Come here!'): print (name,foo)

defaultArg('Akhil')

**Output:-**



1. **Write python functions to use the concept of variable length argument & global variable. Write a program to use these functions**

**Code:-**

result = 100 def sum(\*args):

for arg in args: print(arg)

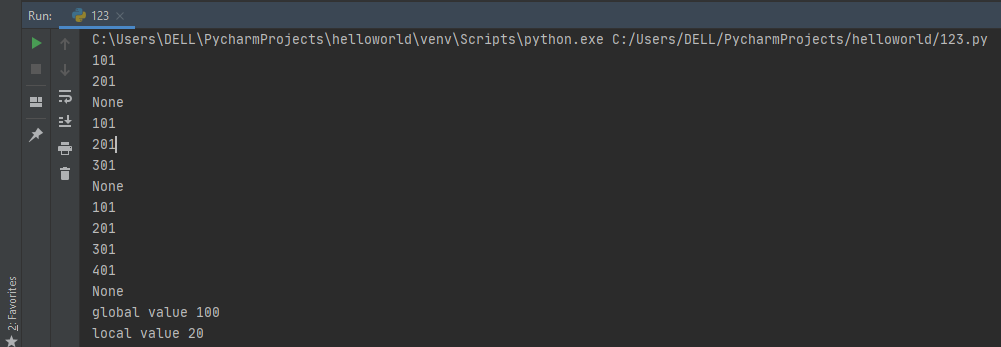
def fun(): result = 20 return result

print(sum(101, 201))

print(sum(101, 201, 301))

print(sum(101, 201, 301, 401))

print("global value", result) print("local value", fun())

**Output:-**

# PROGRAM – 6

**AIM - Program using concept of Class, object, class variable, class method, static method**

1. Create a class Account with name, account no and balance as attribute and no\_of\_accounts as class variable. Account no should be generated automatically (starting from 1) using the class variable no\_of\_account. Add the methods for displaying the account information, depositing given amount, withdrawing given amount and initializer method to initialize the object. Create objects of Account class and call different method to test the class
2. Create a class Employee with name , empid ,salary as attribute and no\_of\_ employee and annual\_incr ( % annual increment) as class variable. empid should be generated automatically (starting from 1) using the class variable, no\_of\_employee. Add the instance methods for displaying the employee information, annually increasing the salary with help of class variable annual\_incr , class method to change the value of annual\_incr and initializer method to initialize the object. Create objects of employee class and call different method to test the class (program using class method)
3. Write a Program to showing the use of built in class attributes ( doc ,

dict , name , module , bases ) and special methods( del ( ),

str ( )) and built in function isinstance()

1. **Create a class Account with name, account no and balance as attribute and no\_of\_accounts as class variable. Account no should be generated automatically (starting from 1) using the class variable no\_of\_account. Add the methods for displaying the account information, depositing given amount, withdrawing given amount and initializer method to initialize the object. Create objects of Account class and call different method to test the class**

**Code:-**

class MyClass:

"\*\*\*\*MY BANKING SYSTEM\*\*\*\*" account\_number=1

def init (self,name,acno,adhaar,balance): self.name=name

self.acno=MyClass.account\_number MyClass.account\_number+=1 self.adhaar=adhaar self.balance=balance

def bank(self): print("Name:",self.name) print("Account number:",self.acno) print("adhaar card No.",self.adhaar) print("balance:",self.balance)

def withdraw(self,amount): if(amount>self.balance):

print("insufficient balance") print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

else:

self.balance-=amount

print("available balance=",self.balance) print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

def deposit(self,amount): self.balance+=amount

print("new available balance:",self.balance) print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

def display(self):

print("\*\*\*\*\*\*ACCOUNT DETAILS\*\*\*\*\*\*\*") print("Name:",self.name)

print("Account number:",self.acno) print("adhaar card No.",self.adhaar) print("balance:",self.balance)

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*") obj1=MyClass("Atul",63,2500,5000000) print(MyClass. doc )

obj1.bank() while(1):

print("Enter operation:") print("1. Withdraw") print("2. Deposit")

print("3. Display A/C details") print("4. Exit") x=(int(input("enter choice: "))) if x==1:

amount=(int(input("enter amount to withdraw:"))) obj1.withdraw(amount)

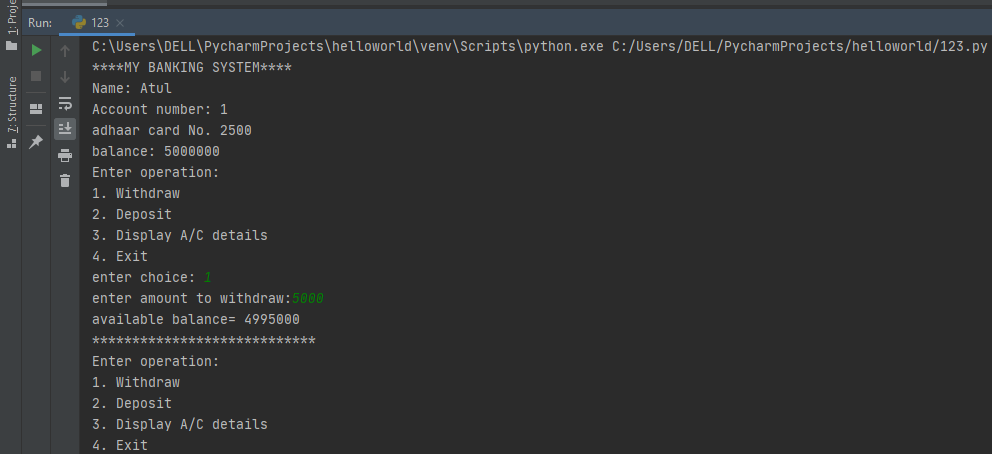
elif x==2:

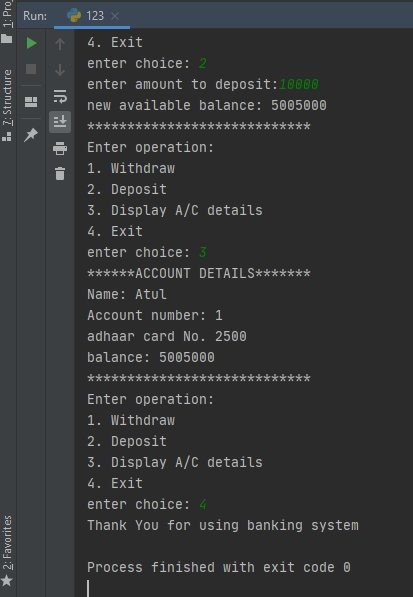
amount=(int(input("enter amount to deposit:"))) obj1.deposit(amount)

elif x==3: obj1.display()

elif x==4:

print("Thank You for using banking system") break

**Output:-**



1. **Create a class Employee with name , empid ,salary as attribute and no\_of\_ employee and annual\_incr ( % annual increment) as class variable. empid should be generated automatically (starting from 1) using the class variable, no\_of\_employee. Add the instance methods for displaying the employee information, annually increasing the salary with help of class variable annual\_incr , class method to change the value of annual\_incr and initializer method to initialize the object. Create objects of employee class and call different method to test the class (program using class method**

**Code:-**

class Employee:

"additional documentation" increment = 1

id\_number = 1

def init (self, name, empid, adhaar, salary): self.name = name

self.empid = Employee.id\_number Employee.id\_number += 1 self.adhaar = adhaar

self.salary = salary

def display(self): print("Name:", self.name)

print("Employee id number:", self.empid) print("adhaar card No.", self.adhaar) print("salary :", self.salary)

@classmethod

def raise\_increment(cls, inc): cls.increment = cls.increment + inc

def raise1(self):

self.salary = self.salary \* Employee.increment return self.salary

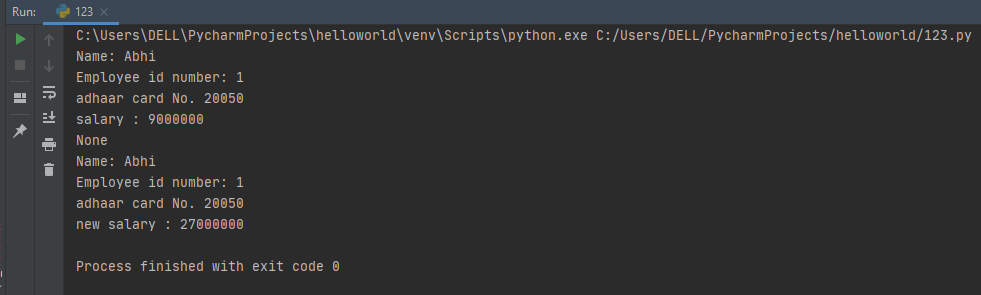
def display1(self): print("Name:", self.name)

print("Employee id number:", self.empid) print("adhaar card No.", self.adhaar) print("new salary :", self.salary)

obj1 = Employee("Abhi", "", 20050, 9000000) obj1.display() print(Employee.raise\_increment(2)) obj1.raise1()

obj1.display1()

**Output:-**



1. **Write a Program to showing the use of built in class attributes ( doc ,**

**dict , name , module , bases ) and special methods( del (**

**), str ( )) and built in function isinstance()**

**Code:-**

class MyClass:

"additinal documentation" b = 5

def init (self):

print("My name is Abhimanyu") print(str('10'))

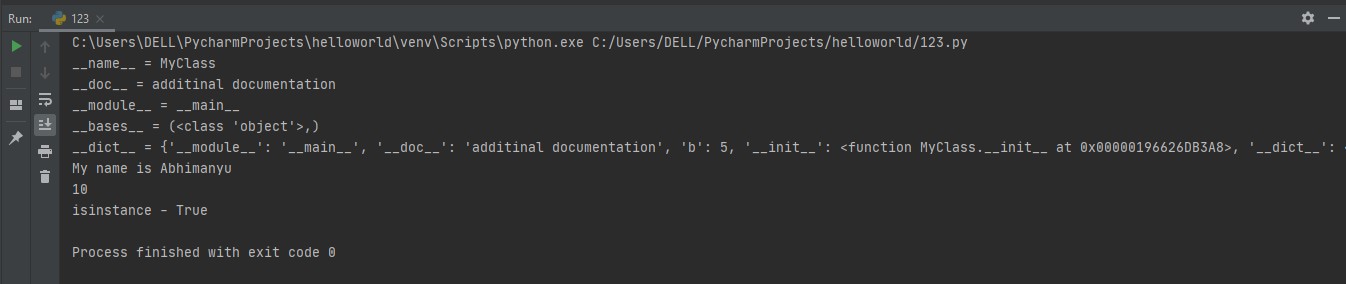
print(" name =", MyClass. name ) print(" doc =", MyClass. doc )

print(" module =", MyClass. module ) print(" bases =", MyClass. bases ) print(" dict =", MyClass. dict )

a = MyClass()

print("isinstance -", isinstance(a, MyClass)) del a

**Output:-**



# PROGRAM- 7

**a) AIM- Program using the concept of Inheritance**

1. Create a class Polygon to represent a polygon having no of sides and a list having magnitude of each side as attribute. Add the inputSides() to input sides and displaySides() to display sides as methods. Derive a class Triangle from Polygon and add an additional method displayArea() to display area. Create object of Triangle and call different methods to test the class
2. Create a class Person having name, age, as attributes, init () method to initialize the object and display() to display person information. Derive a class Student from Person having roll no, University name, branch as additional attributes and init (), display() to display student information and change\_Branch() method. Create object of Student type and call different methods to test the class.
3. Write a program to show the concept of multiple inheritance in python
4. **Create a class Polygon to represent a polygon having no of sides and a list having magnitude of each side as attribute. Add the inputSides() to input sides and displaySides() to display sides as methods. Derive a class Triangle from Polygon and add an additional method displayArea() to display area. Create object of Triangle and call different methods to test the class**

**Code:-**

class Polygon:

def init (self, no\_of\_sides): self.n = no\_of\_sides

self.sides = [0 for i in range(no\_of\_sides)] def inputSides(self):

self.sides = [float(input("Enter side "+str(i+1)+" : ")) for i in range(self.n)] def dispSides(self):

for i in range(self.n): print("Side",i+1,"is",self.sides[i])

class Triangle(Polygon): def init (self):

Polygon. init (self,3) def findArea(self):

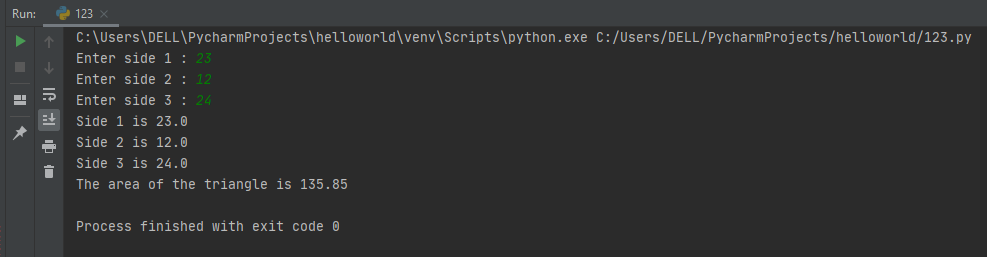
a, b, c = self.sides s = (a + b + c) / 2

area = (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

print('The area of the triangle is %0.2f' %area) obj2=Triangle()

obj2.inputSides() obj2.dispSides() obj2.findArea()

# Output:-



1. **Create a class Person having name, age, as attributes, init () method to initialize the object and display() to display person information. Derive a class Student from Person having roll no, University name, branch as additional attributes and init (), display() to display student information and change\_Branch() method. Create object of Student type and call different methods to test the class.Code:-**

class Person:

"optional documentation"

def init (self,name,age,adhaar): self.name=name;

self.age=age; self.adhaar=adhaar;

def display1(self): print("Name:",self.name) print("employee id:",self.age) print("adhaar card No.",self.adhaar)

class Student (Person):

"student class inherits person"

def init (self,name,age,adhaar,rollno,branch): super(). init (name,age,adhaar) self.rollno=rollno

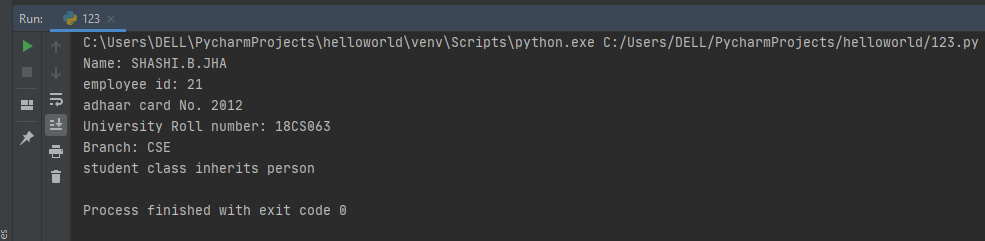
self.branch=branch def display2(self):

print("University Roll number:",self.rollno) print("Branch:",self.branch)

obj1=Student("SHASHI.B.JHA",21,2012,"18CS063","CSE")

obj1.display1() obj1.display2() print(Student. doc )

**Output:-**



1. **Write a program to show the concept of multiple inheritance in python Code:-**

class Person:

def init (self, personName, personAge): self.name = personName

self.age = personAge def showName(self):

print(self.name) def showAge(self):

print(self.age) class Student:

def init (self, studentId): self.studentId = studentId

def getId(self):

return self.studentId

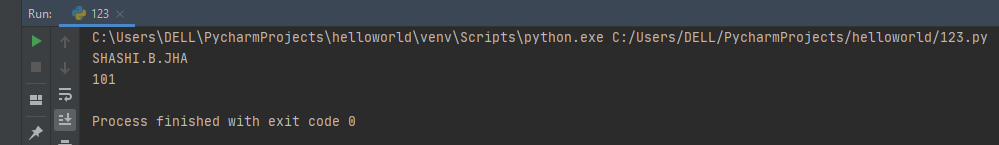
class Resident(Person, Student): def init (self, name, age, id):

Person. init (self, name, age) Student. init (self, id)

resident1 = Resident('SHASHI.B.JHA', 21, '101') resident1.showName()

print(resident1.getId())

**Output:-**



# PROGRAM-8

**AIM- Program using the concept of Polymorphism, operator overloading**

1. In a retail outlet there are two modes of bill Payment (1) Cash : Calculation includes VAT(10%) Total Amount = Purchase amount + VAT (2) Credit card: Calculation includes processing charge and VAT Total Amount = Purchase amount + VAT (10%) + Processing charge (2%) The act of bill payment is same but the formula used for calculation of total amount diﬀers as per the mode of payment. Can the Payment maker simply call a method and that method dynamically selects the formula for the total amount? Demonstrate this Polymorphic behaviour with code.
2. Write a program to create a class to represent length in feet and inch. Overload the “+” operator to add the two object of length type.
3. Write a program to overload comparison operator in python
4. **In a retail outlet there are two modes of bill Payment (1) Cash : Calculation includes VAT(10%) Total Amount = Purchase amount + VAT**

**(2) Credit card: Calculation includes processing charge and VAT Total Amount = Purchase amount + VAT (10%) + Processing charge (2%) The act of bill payment is**

**same but the formula used for calculation of total amount differs as per the mode of payment. Can the Payment maker simply call a method and that method dynamically selects the formula for the total amount? Demonstrate this Polymorphic behaviour with code.**

**CODE: -**

from abc import ABC, abstractmethod class payment(ABC):

VAT =1.10

@abstractmethod

def calTotal(self,purchasedAmount): pass

class creditCard(payment): processingCharge=1.02

def calTotal(self,purchasedAmount): amt=purchasedAmount\* payment.VAT amt=amt \*self.processingCharge

print("VAT: ", payment.VAT,"\nprocessing fee= ",self.processingCharge) return(amt)

class cashPayment(payment):

def calTotal(self,purchasedAmount): amt=purchasedAmount\* payment.VAT print("VAT: ", payment.VAT) return(amt)

class bill():

def init (self,purchasedAmount): self.purchasedAmount= purchasedAmount

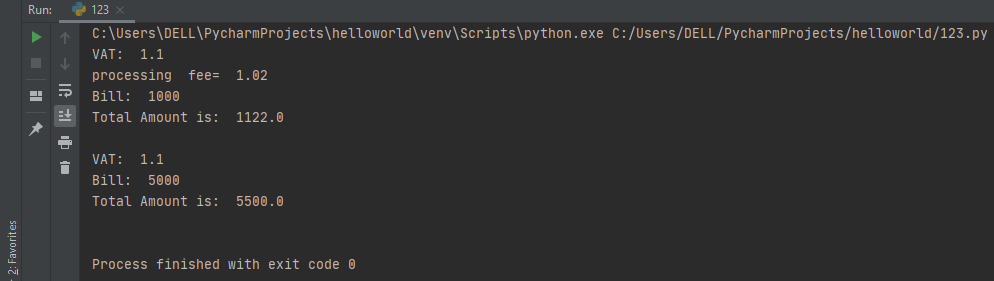
def makePayment(self,mode): if isinstance(mode, payment):

totalAmount= mode.calTotal(self.purchasedAmount)

print("Bill: ",self.purchasedAmount,"\nTotal Amount is: ",totalAmount,"\n")

b1=bill(1000) b2=bill(5000) credit\_card= creditCard() cash= cashPayment()

b1.makePayment(credit\_card) b2.makePayment(cash)

**Output:-**

1. **Write a program to create a class to represent length in feet and inch. Overload the “+” operator to add the two object of length type.**

**Code:-**

class length:

def init (self, feet=0, inch=0): self.feet = feet

self.inch = inch

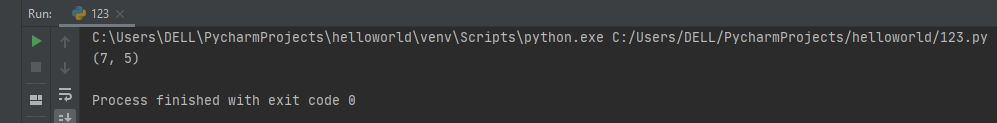
def add (self, other):

inch = (self.inch + other.inch) % 12

feet = (self.feet + other.feet) + (self.inch + other.inch) // 12 return (feet, inch)

l1 = length(3, 9) l2 = length(3, 8) l3 = l1 + l2 print(l3)

**Output:-**



1. **Write a program to overload comparison operator in python**

**Code:-**

class distance:

def init (self, x=5, y=5): self.ft = x

self.inch = y

def eq (self, other):

if self.ft == other.ft and self.inch == other.inch: return "both objects are equal"

else:

return "both objects are not equal"

def ge (self, other):

in1 = self.ft \* 12 + self.inch in2 = other.ft \* 12 + other.inch if in1 >= in2:

return "first object greater than or equal to other" else:

return "first object smaller than other"

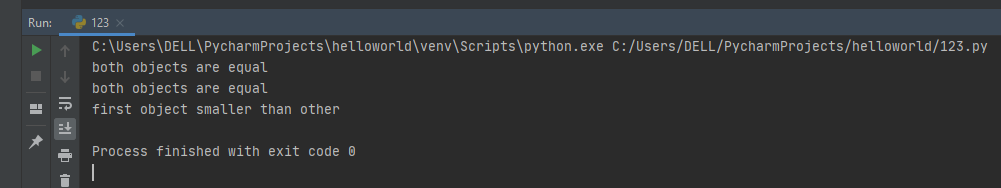
d1 = distance(5, 5) d2 = distance() print(d1 == d2)

d3 = distance()

d4 = distance(6, 10) print(d1 == d2)

d5 = distance(3, 11) d6 = distance() print(d5 >= d6)

**Output:-**



# PROGRAM-9

**AIM - Program on file handling in Python**

1. Write a python program to write few lines on a file, read it back and create a dictionary having each word in file as keys in dictionary and occurrence of these word as values and print the dictionary.
2. A file student.txt store student information. Information about each student is written on separate line in the form: roll-no student-name (student-name may consist of any number of words).Write a Python program that takes student roll no as input and print the student name. If roll no is not present in the file it display : “roll no not present in the file”
3. Write a python program to read a file that contains email ids on the separate lines in the form: “[personname@companyname.com.](mailto:personname@companyname.com) Create a new file that contain only company names, read the new file to print the company name
4. **Write a python program to write few lines on a file, read it back and create a dictionary having each word in file as keys in dictionary and occurrence of these word as values and print the dictionary.**

**Code:-**

with open ("C:/Users/DELL/Documents/abc.txt","r") as f: line =f.read().replace("\n","")

word = line.split(" ") dict1={}

for i in range (0,len(word)): ch=word.pop()

if(ch==" " or ch=="," or ch==";" or ch==":"): continue

elif ch not in dict1: dict1[ch] =1

else:

dict1[ch]+=1 print(dict1)

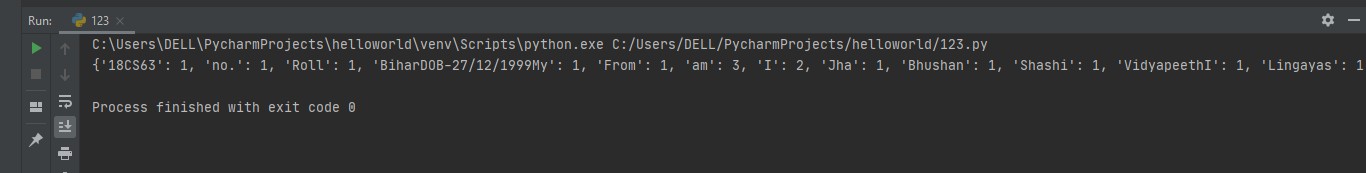
**TEXT FILE:-**

I am Persuing Btech from Lingayas Vidyapeeth I am Shashi Bhushan Jha

I am From Bihar DOB-27/12/1999

My Roll no. 18CS63

**Output:-**



1. **A file student.txt store student information. Information about each student is written on separate line in the form: roll-no student-name (student-name may consist of any number of words).Write a Python program that takes student roll no as input and print the student name. If roll no is not present in the file it display : “roll no not present in the file”**

**Code:-**

def stud():

with open ("C:/Users/DELL/Documents/1.txt","r") as f: line =f.read().replace("\n"," ")

student\_list = line.split(" ")

a=input("Enter the roll-no.: ")

if a in student\_list:

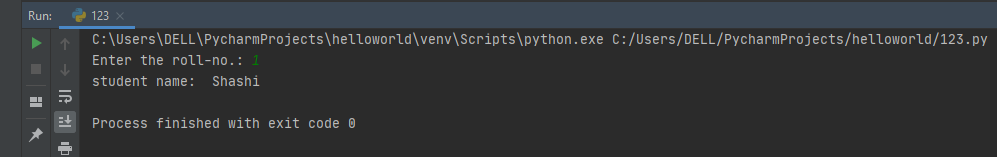
print("student name: ", student\_list[student\_list.index(a)+1]) else:

print("Roll-number not present")

stud()

**TEXT FILE:-**

1. Shashi
2. Abhi
3. Arjun
4. Akhil

**Output:-**

1. **Write a python program to read a file that contains email ids on the separate lines in the form:** [**“per**](mailto:personname@companyname.com)**s**[**onname@companyname.com.**](mailto:personname@companyname.com) **Create a new file that contain only company names, read the new file to print the company name**

**Code:-**

def CompanyAdd():

with open ("C:/Users/DELL/Documents/email.txt","r") as f: username =f.read().replace(".com","")

company = username.split("\n") f.close()

file\_list=""

for i in company:

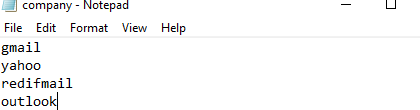
n = i.index("@") file\_list+=i.replace(i[0:n+1],"") file\_list+="\n"

with open ("C:/Users/DELL/Documents/company.txt","w") as f1: f1.write(file\_list)

f1.close() CompanyAdd()

**TEXT FILE:-**

[shashi@gmail.com](mailto:shashi@gmail.com) [hello@yahoo.com](mailto:hello@yahoo.com) [python@redifmail.com](mailto:python@redifmail.com) [user@outlook.com](mailto:user@outlook.com) **Output:-**



# PROGRAM -10

**AIM: Program on Exception handling**

1. Write a function divide (arg1, arg2) to divide arg1 by arg2. Use the exception handling mechanism to handle all type of possible exceptions that may occur. Take the value of arg1 and arg2( of any type) from user as input and call the function divide to print the result of division or suitable message if any type of exception occurs( use also else and finally block)
2. Write a program to open a file in read only mode read data from file and then try to write data on file. Use the exception handling mechanism to handle all type of possible exception
3. **Write a function divide (arg1, arg2) to divide arg1 by arg2. Use the exception handling mechanism to handle all type of possible exceptions that may occur. Take the value of arg1 and arg2( of any type) from user as input and call the function divide to print the result of division or suitable message if any type of exception occurs( use also else and finally block)**

**Code:-**

result = 0

def divide(x,y):

return(float(x)/y)

try:

x=input("Enter the first number: ") y=input("Enter the second number: ") divide(x,y)

except ZeroDivisionError:

print ("Denominator can't be zero!\n'ZeroDivisionError'") except SyntaxError:

print ("Don't use blank spaces\n'SyntaxError'!") except NameError:

print ("Wrond input type!Enter integers only!\n'NameError'")

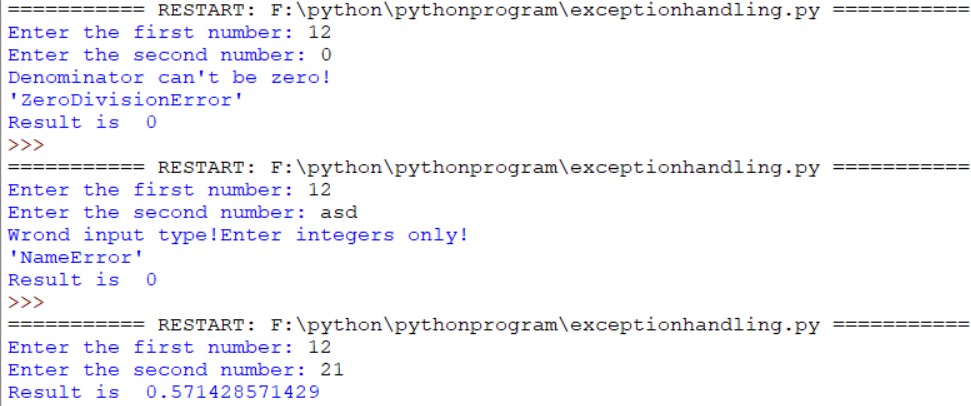
else:

result = divide(x,y)

finally:

print ("Result is ",result)

**Output:-**



1. **Write a program to open a file in read only mode read data from file and then try to write data on file. Use the exception handling mechanism to handle all type of possible exception**

**Code:-**

def fileOpen():

with open("sample.txt", "r") as f1: readFile = f1.read()

newfile = readFile.replace(".", "\n") f1.write(newfile)

f1.close()

try:

fileOpen()

except IOError as e: print(e, ":IOError")

**Output:**