Python and Django Internship Report

Personal Details

Name	:	Patel Priya .P
College Name	:	Ss Agrawal navsari
Degree	:	BE
Semester	:	7
Github URL	:	https://github.com/piyu

Company Details

Company Name	:	Akash Technolabs
External Guide	:	Akash Padhiyar
Training Duration	:	26-05-2021 to 21-06-2021

Index

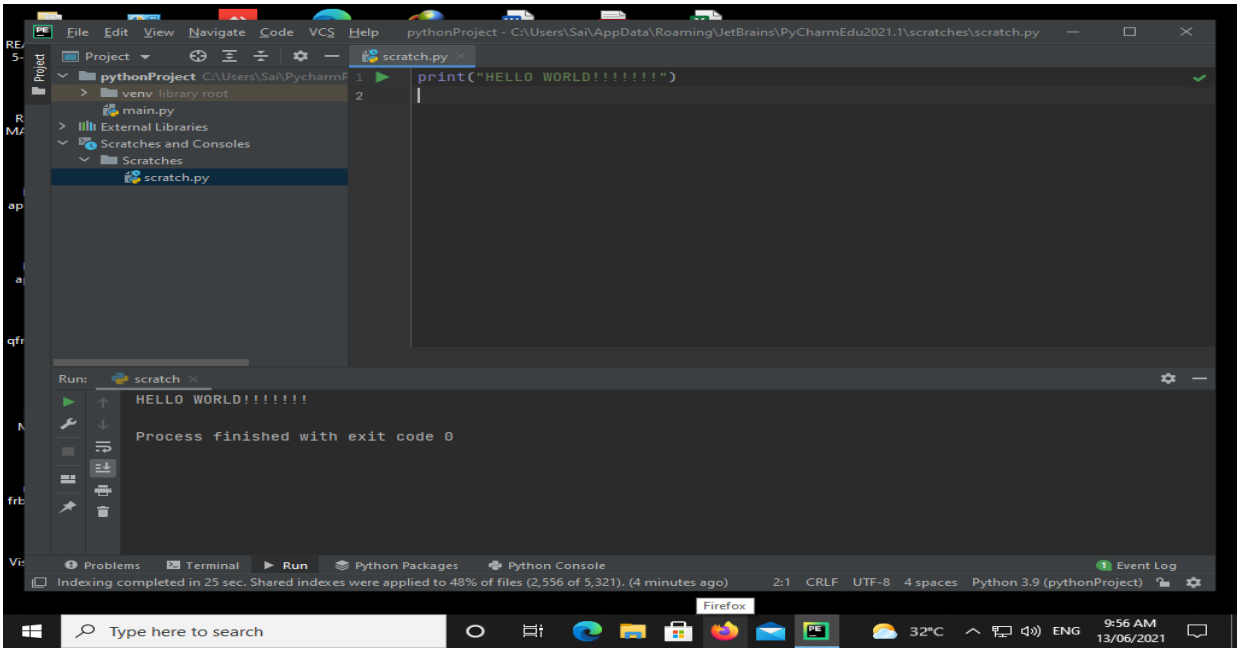
Task:1

Print hello world in python:-

Code:-

```
print("HELLO WORLD!!!!!!!")
```

Output:-



Task:2

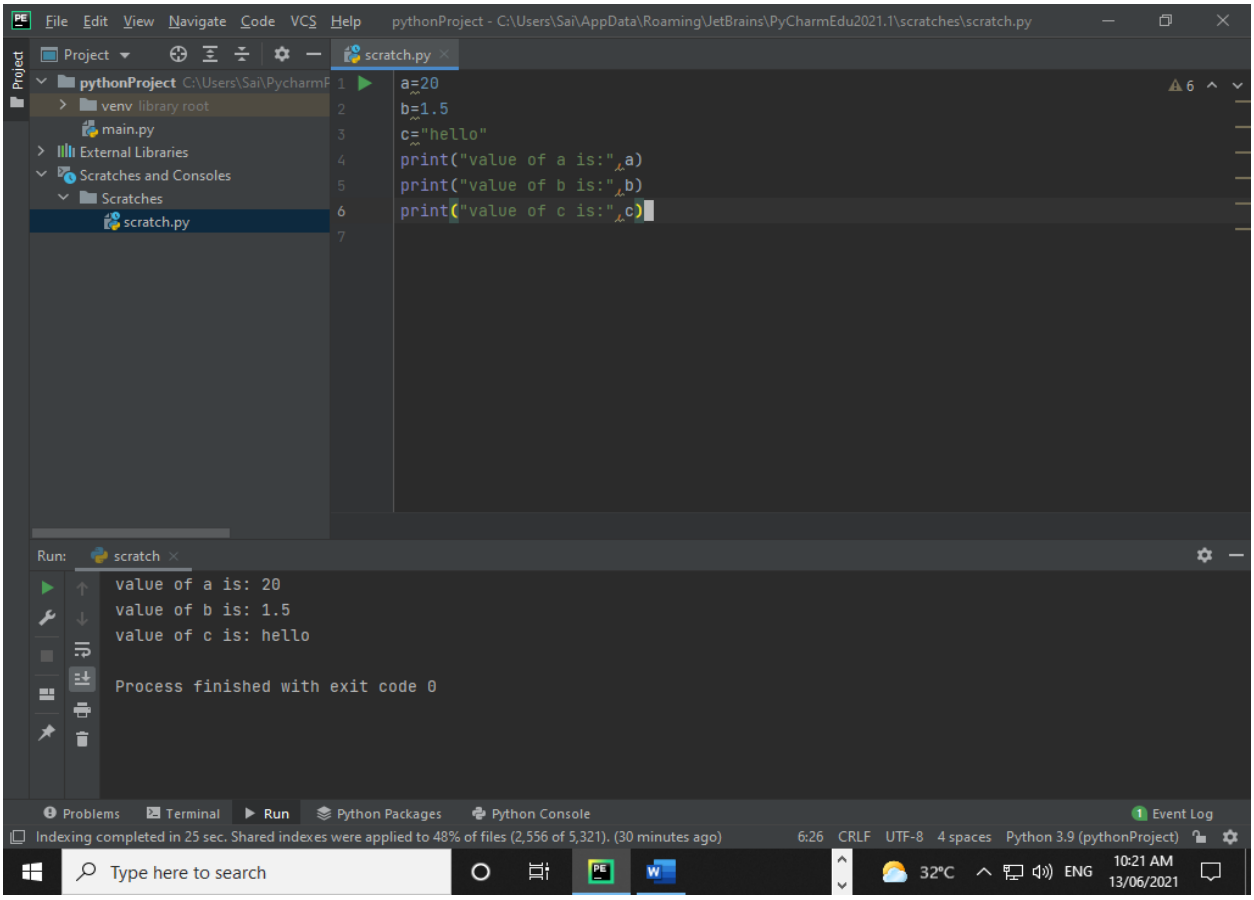
Variable and Datatype:-

Variable:- Variable is named location used to store data in memory.

Code:-

```
a=20
b=1.5
c="hello"
print("value of a is:",a)
print("value of a is:",b)
print("value of a is:",c)
```

Output:-

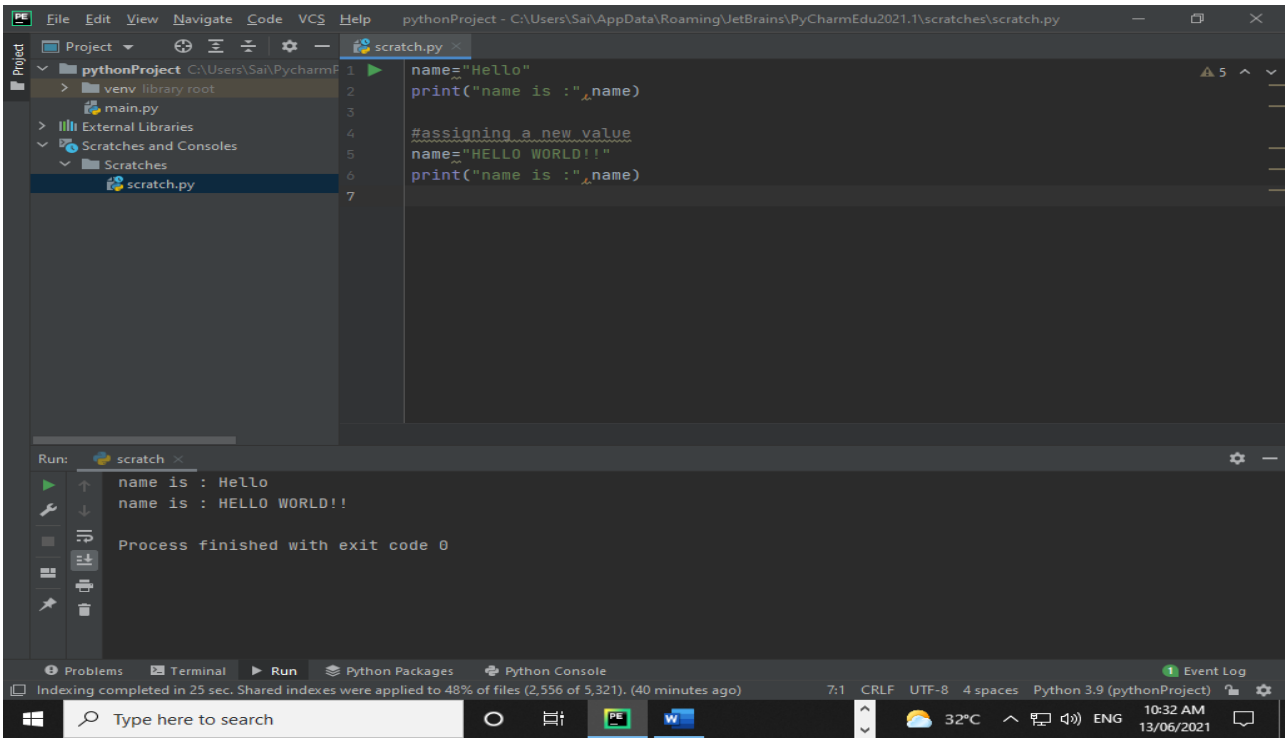


Changing the value of variable

Code:-

```
name="Hello"
print("name is :",name)

#assigning a new value
name="HELLO WORLD!!"
print("name is :",name)
```

Output:-

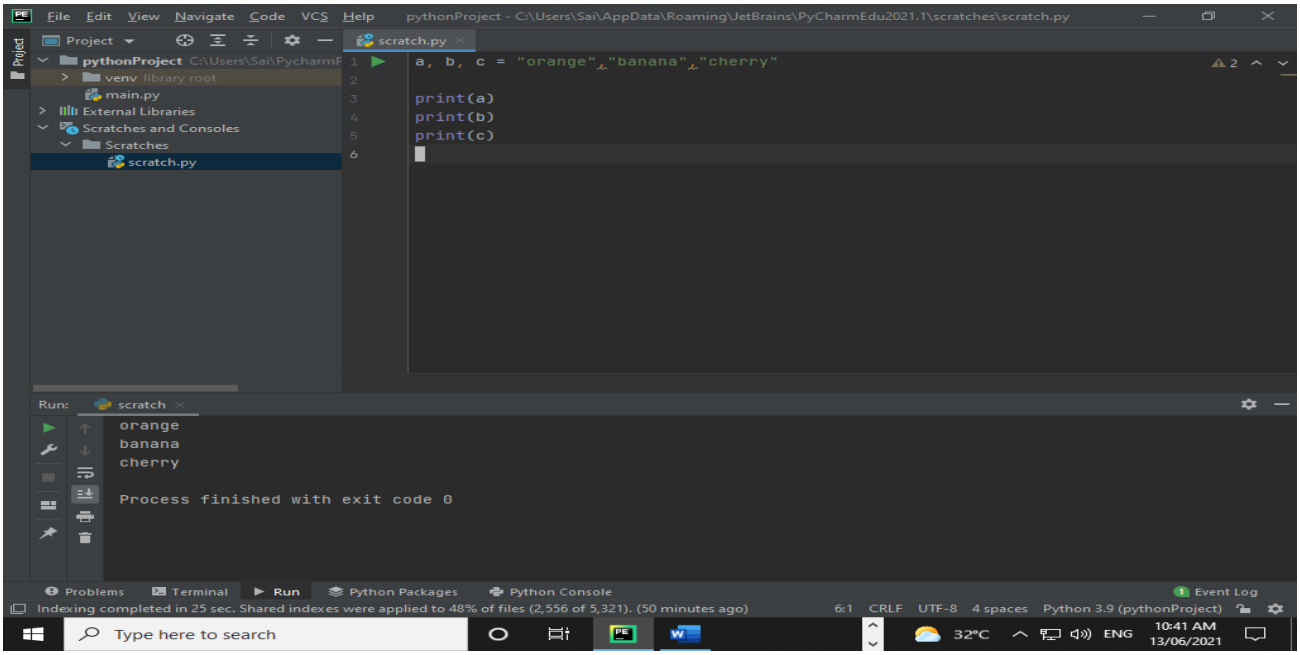


Assign multiple values:-

Code

```
a, b, c = "orange", "banana", "cherry"

print(a)
print(b)
print(c)
```

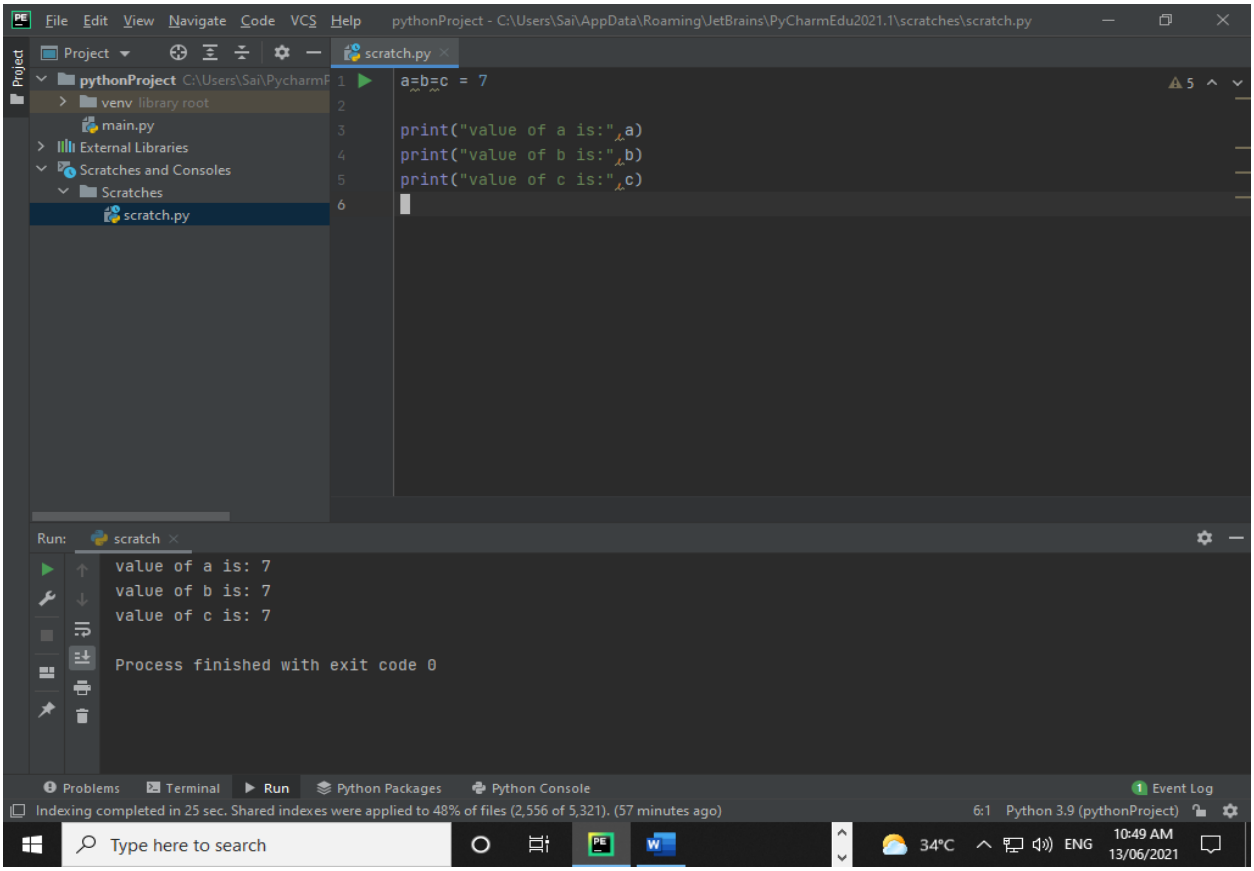
Output:-



Assign the same value to multiple variables

```
a=b=c = 7

print("value of a is:",a)
print("value of b is:",b)
print("value of c is:",c)
```

Output:-



Data type:-

Python has 5 standard data types namely:-

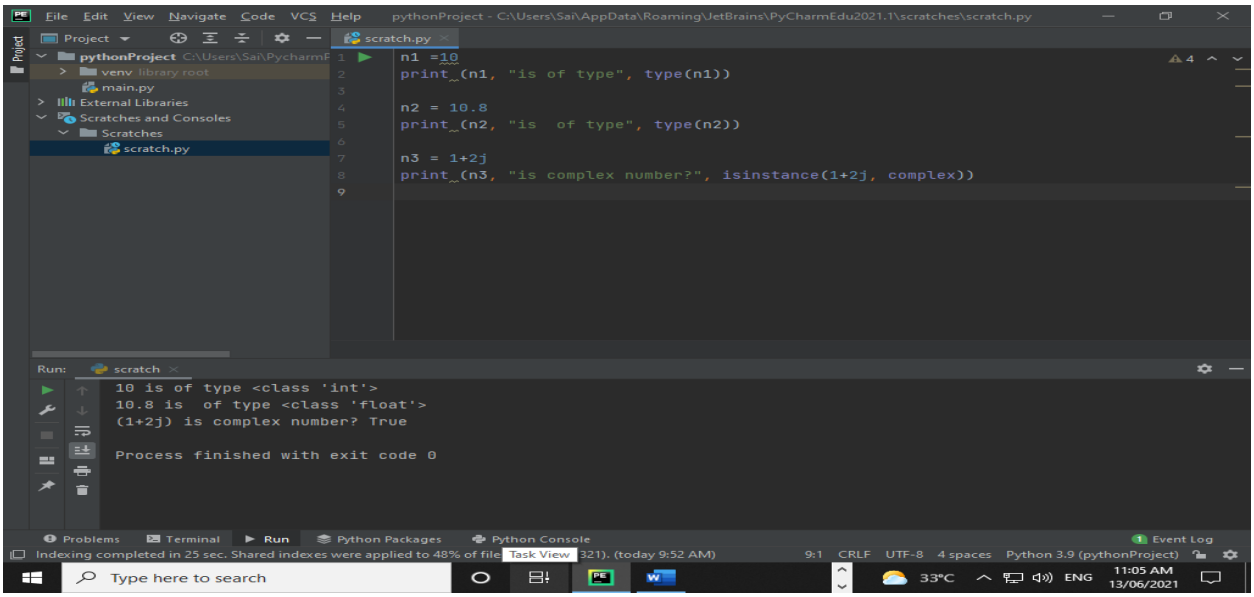
- Numbers
- String
- List
- Tuple
- Dictionary
- Boolean- true and false

Number:-

Code:

```
n1 =10
print (n1, "is of type", type(n1))

n2 = 10.8
print (n2, "is  of type", type(n2))

n3 = 1+2j
print (n3, "is complex number?", isinstance(1+2j, complex))
```

Output:-

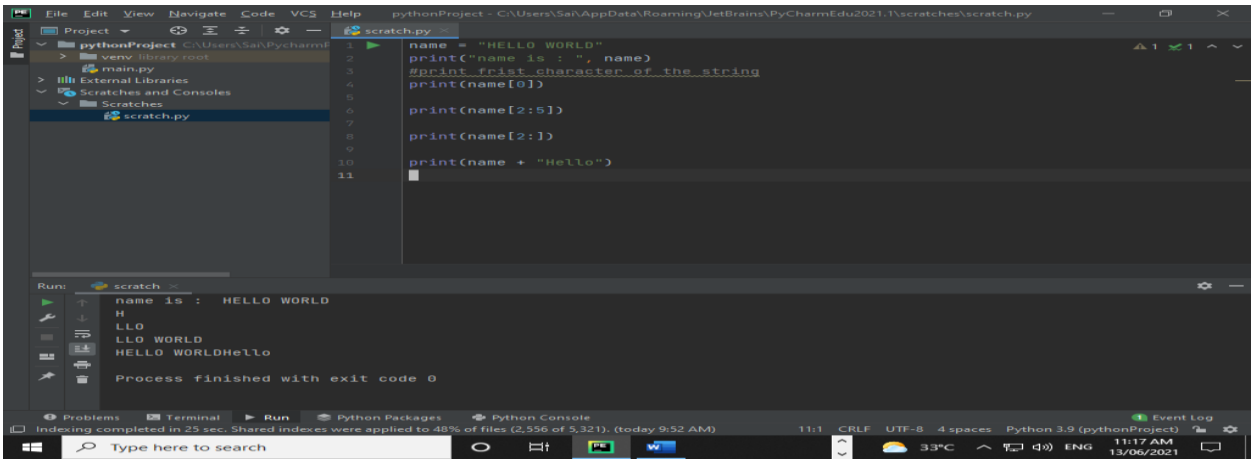


String:-

Code:-

```
name = "HELLO WORLD"
print("name is : ", name)
#print frist character of the string
print(name[0])

print(name[2:5])

print(name[2:])

print(name + "Hello")
```

Output:-

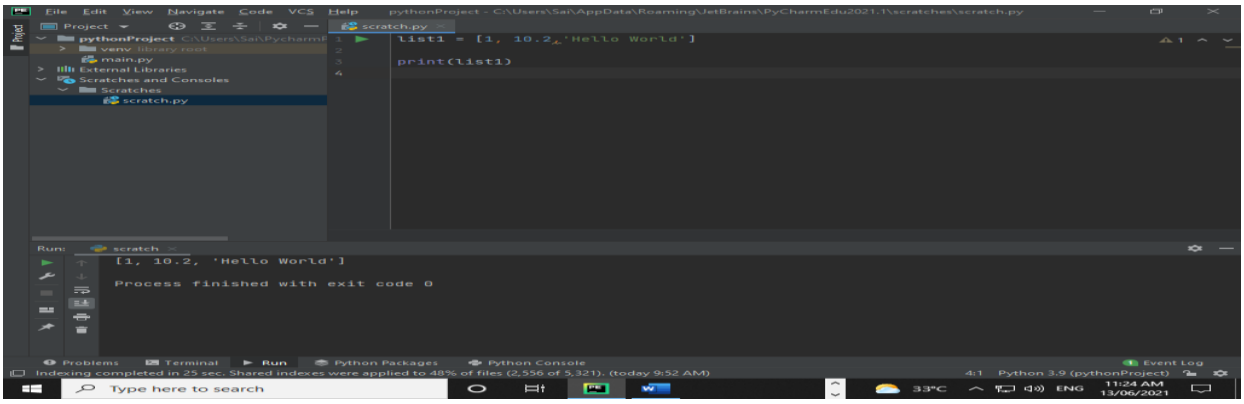


List:-

Code:

```
list1 = [1, 10.2,'Hello World']

print(list1)
```

Output:-

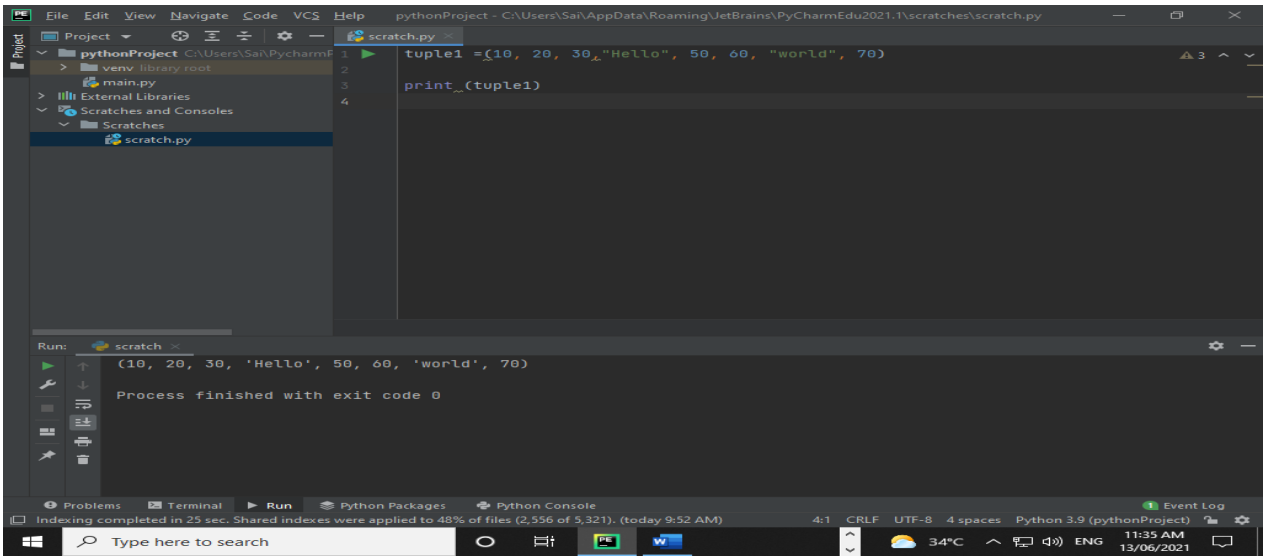


Tuple:-

Code:

```
tuple1 =(10, 20, 30,"Hello", 50, 60, "world", 70)

print (tuple1)
```

Output:-

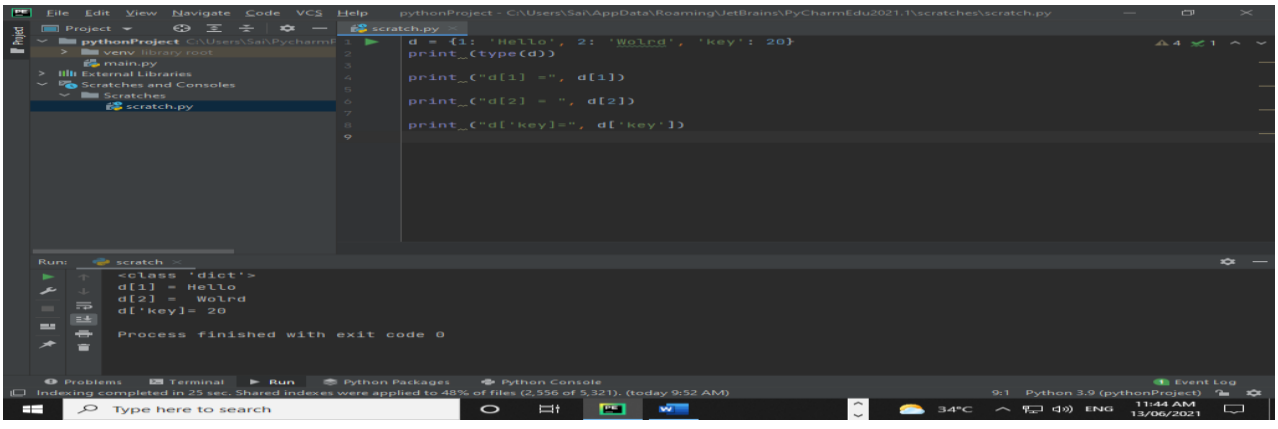


Dictionary:-

Code:

```
d = {1: 'Hello', 2: 'Wolrd', 'key': 20}
print (type(d))

print ("d[1] =", d[1])

print ("d[2] = ", d[2])

print ("d['key]=", d['key'])
```

Output:-



Task:-3

I/o function conditional statement

- Input()function is used to receive input from the console
- **Syntax:-**

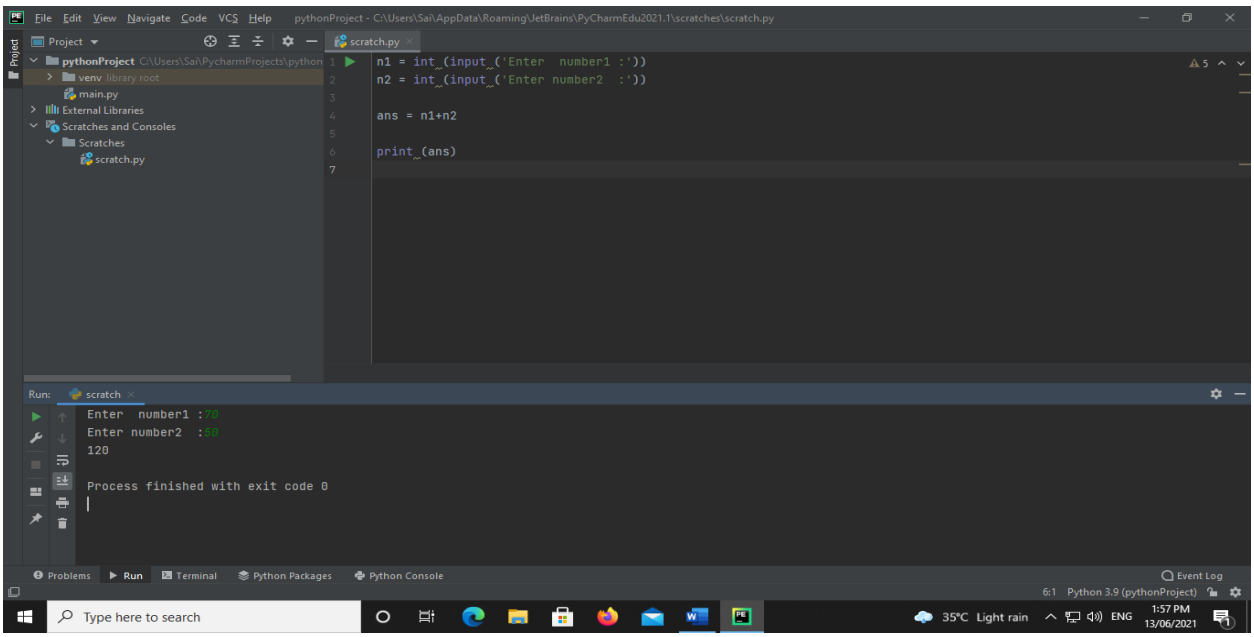
  Input([prompt])

Input:-

Code:

```
n1 = int (input ('Enter  number1 :'))
n2 = int (input ('Enter number2  :'))

ans = n1+n2

print (ans)
```

Output:-



Conditional Statement:

If ….. Statement

Syntax:-

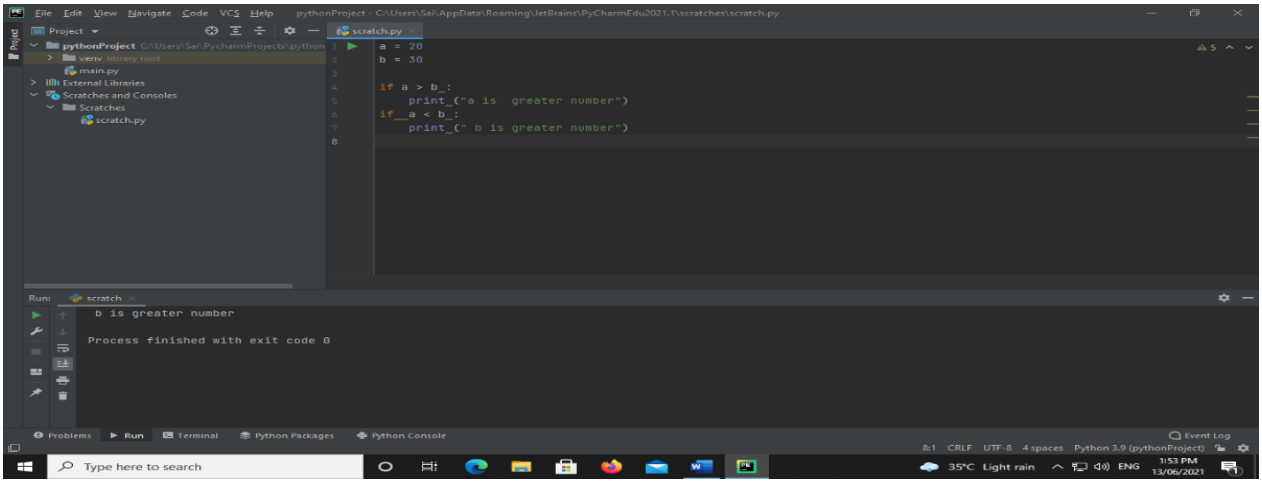
If condition:

Statement(s)

Code:

```
a = 20
b = 30

if a > b :
    print ("a is  greater number")
if  a < b :
    print (" b is greater number")
```

Output:-



If …….else Statement

Syntax:-

If condition:

Statements to be executed

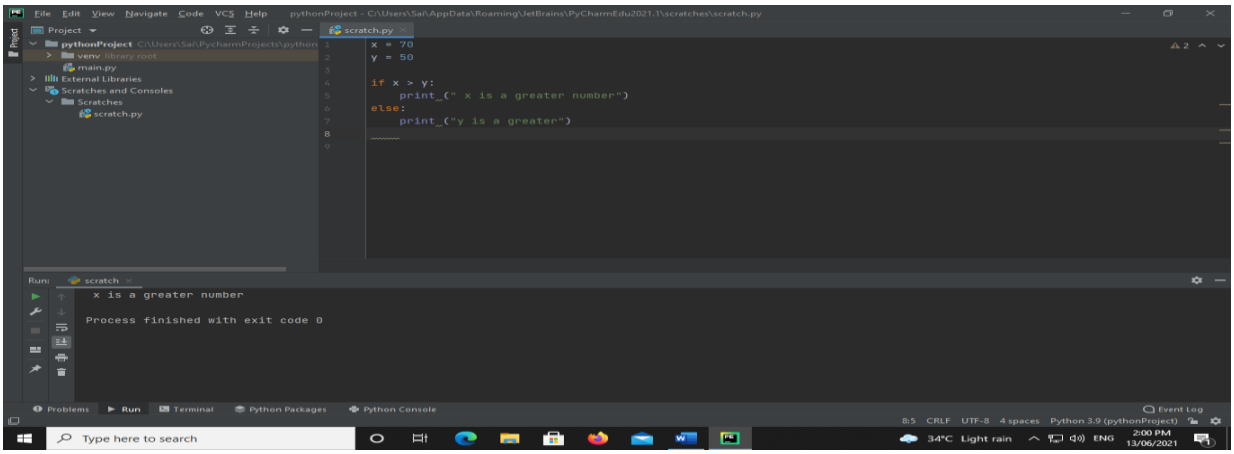
Else:

Statements to be executed

Code:-

```
x = 70
y = 50

if x > y:
    print (" x is a greater number")
else:
    print ("y is a greater")
```

Output:-



If…. elif …else Statement

Syntax:-

If condition:

    Statements to be executed

elif:

    Statements to be executed

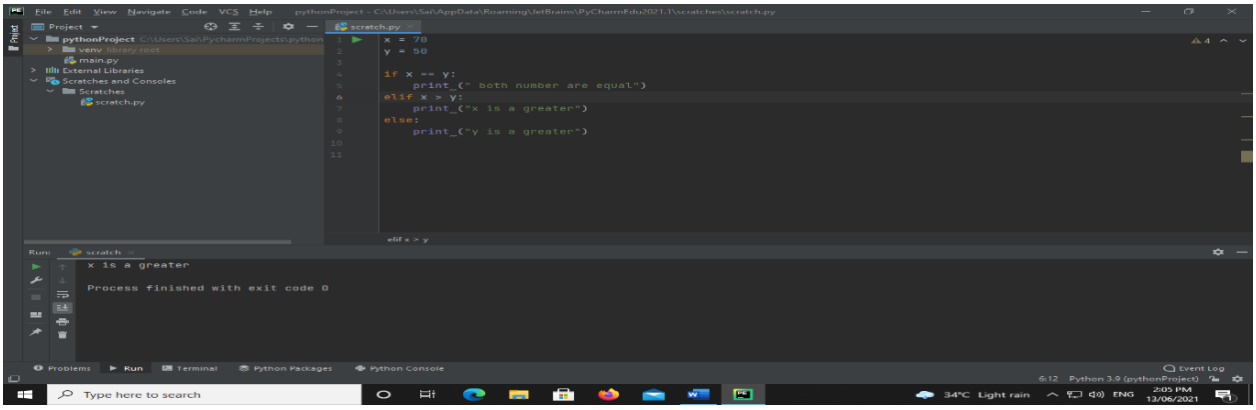
else:

    Statements to be executed

Code:

```
x = 70
y = 50

if x == y:
    print (" both number are equal")
elif x > y:
    print ("x is a greater")
else:
    print ("y is a greater")
```

Output:-

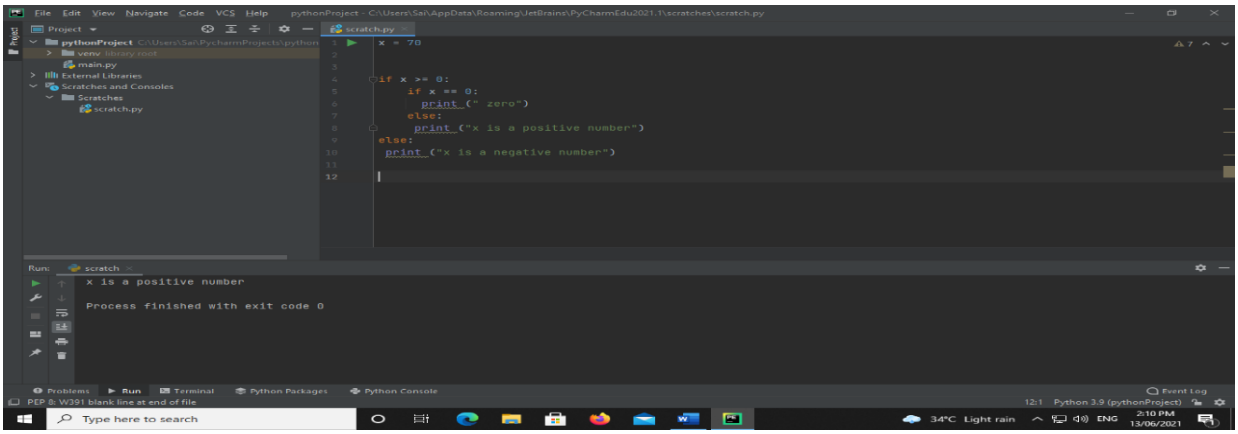


Nested if statement

Code:

```
x = 70


if x >= 0:
    if x == 0:
      print (" zero")
    else:
     print ("x is a positive number")
else:
 print ("x is a negative number")
```

Output:-



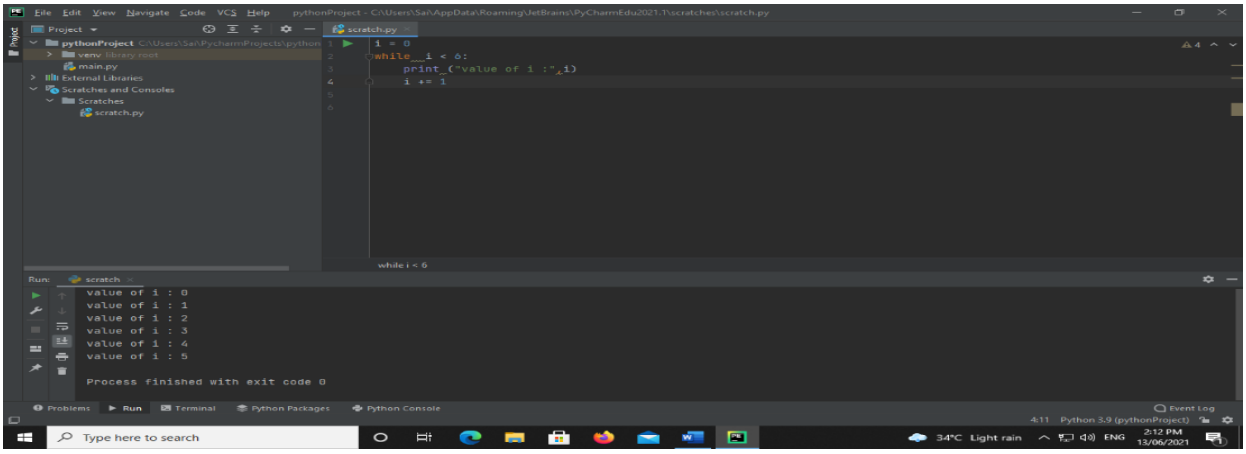
Loop

while loop

code:

```
i = 0
while  i < 6:
    print ("value of i :",i)
    i += 1
```

Output:-

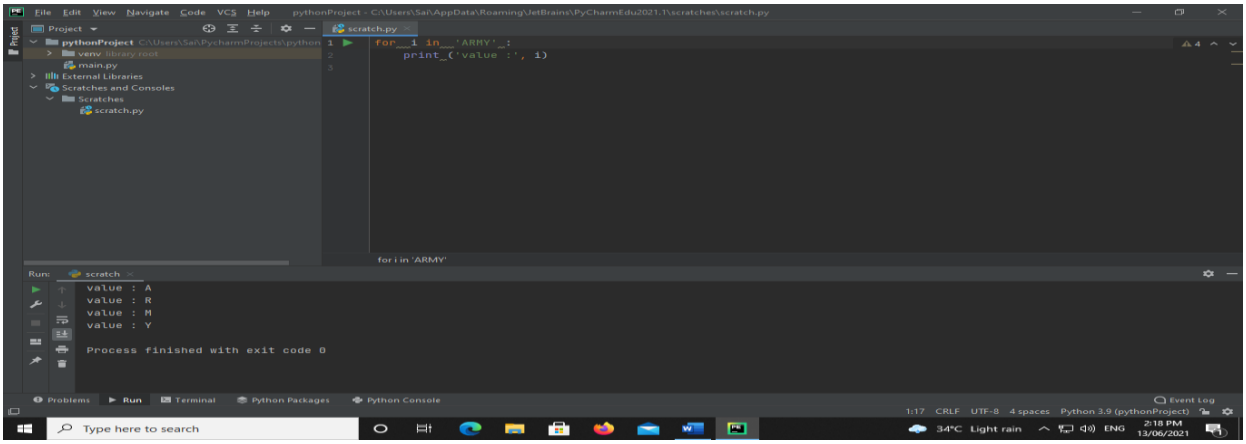


for loop

Code:

```
for i in 'ARMY' :
    print ('value :', i)
```

Output:-

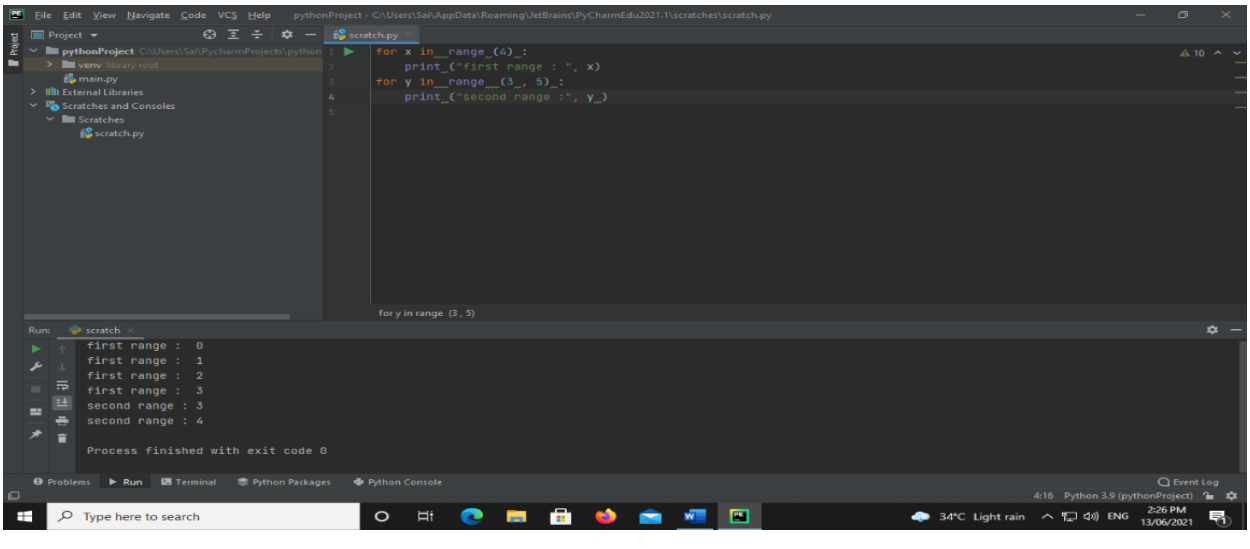


the range () function

Code:

```
for x in  range (4) :
    print ("first range : ", x)
for y in  range  (3 , 5) :
    print ("second range :", y )
```

Output:-



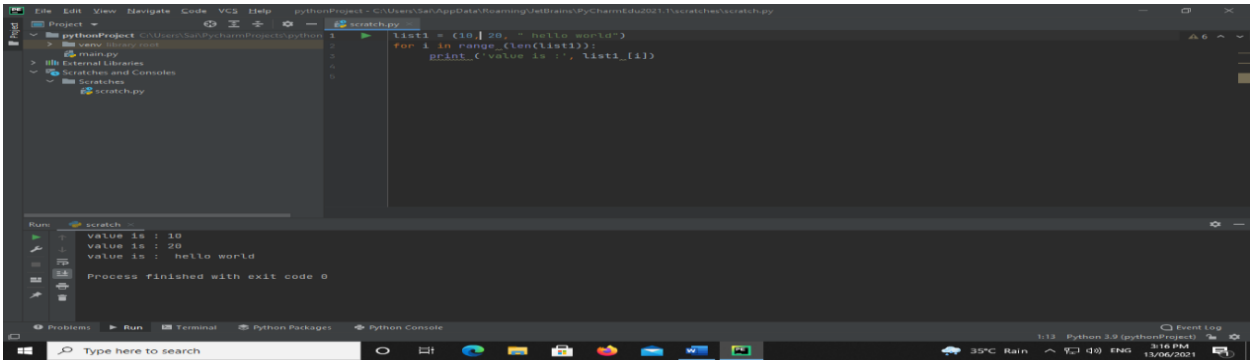
loop with else

Code:

```
list1 = (10, 20, " hello world")
for i in range (len(list1)):
     print ('value is :', list1 [i])
```

```
else:
    print ("no elements")
```

Output:-

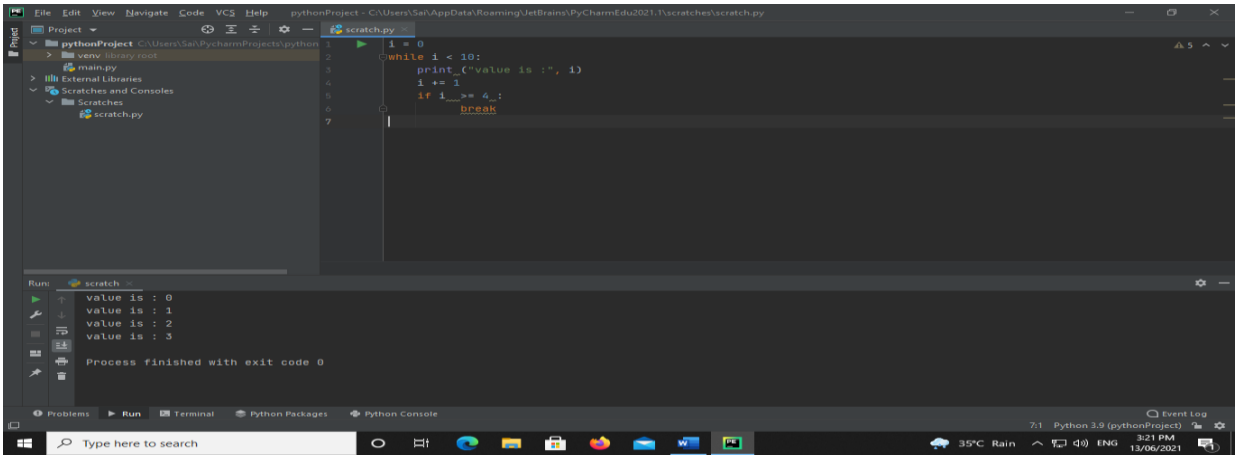


break statement

Code:

```
i = 0
while i < 10:
    print ("value is :", i)
    i += 1
    if i  >= 4 :
        break
```

Output:-



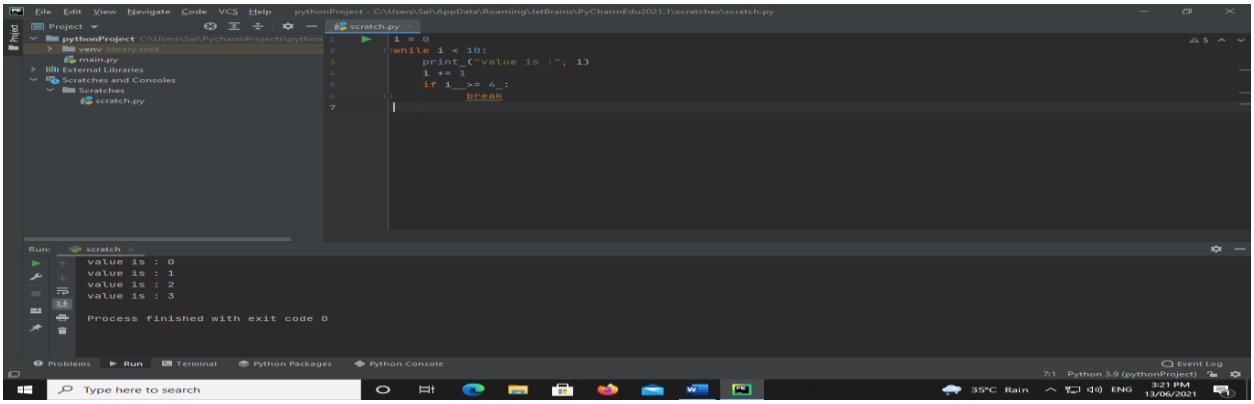
Continue

Code:

```
for x in range (5) :
    if x % 2 ==  0:
```

```
        continue
    print("value is :", x)
```

Output:-

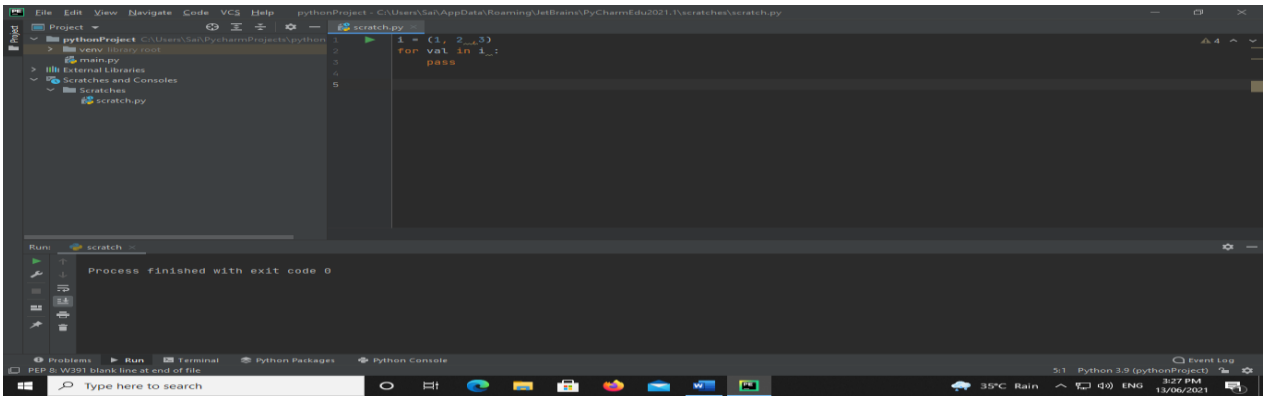


Pass

Code:

```
i = (1, 2 ,3)
for val in i :
    pass
```

Output:-



Task:4

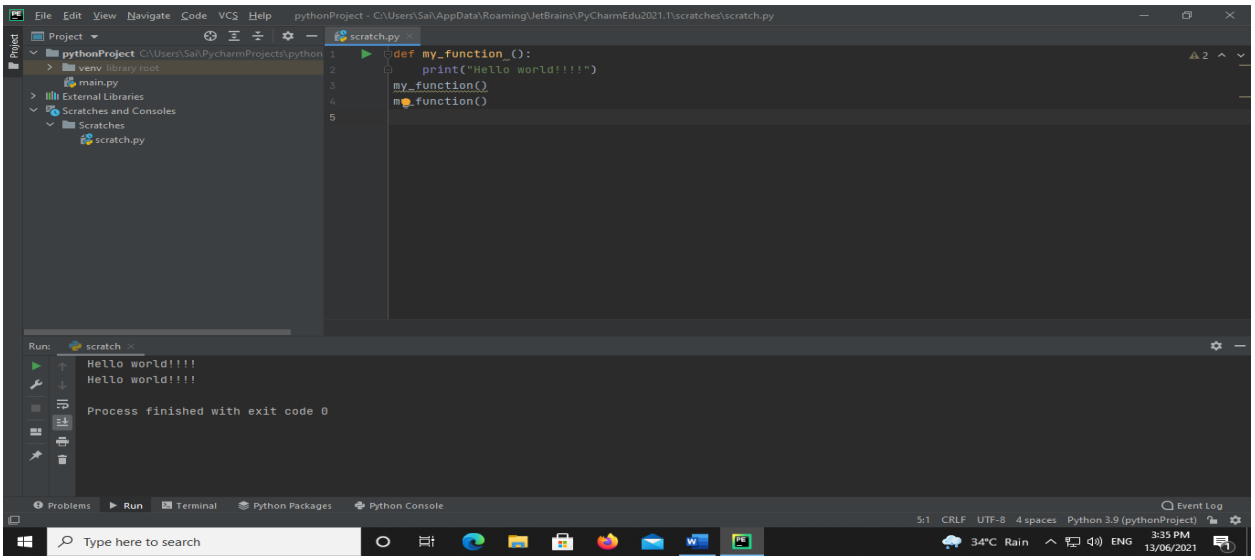
Function in python

- Function is group of related statement that perform a specific task.

Code:

```
def my_function ():
    print("Hello world!!!!")
my_function()
my_function()
```

Output:-

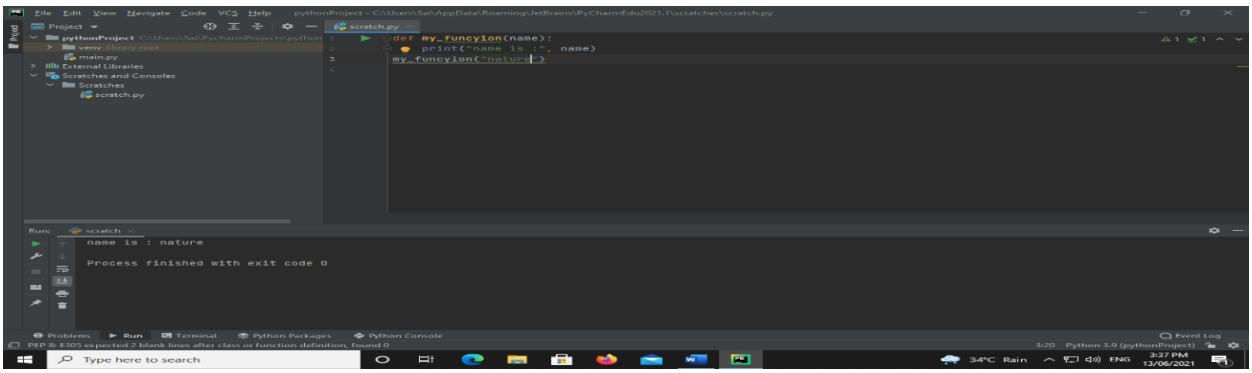


Function with argument:-

Code:-

```
def my_funcyion(name):
    print("name is :", name)
my_funcyion("nature")
```

Output:-

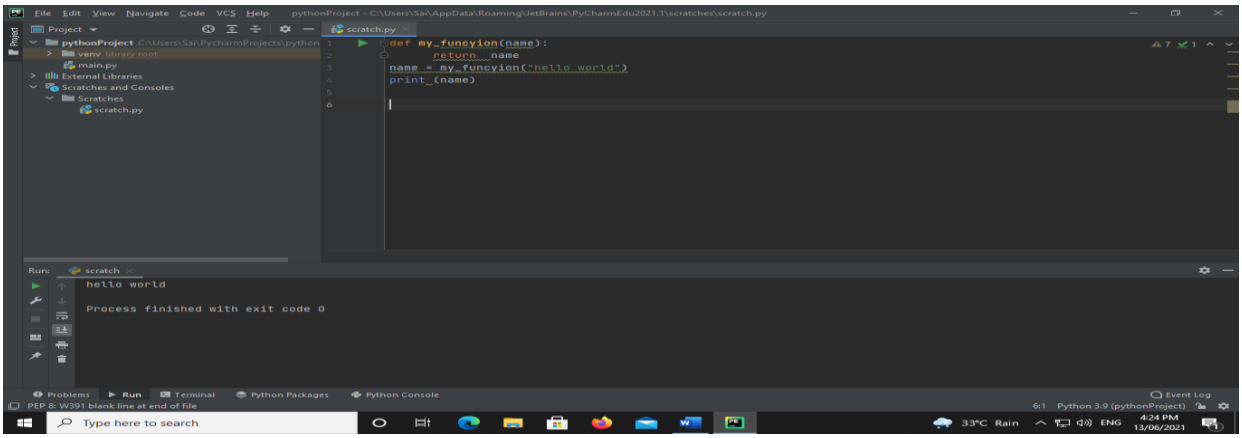


Function with return Statement

Code:

```
def my_funcyion(name):
      return  name
name = my_funcyion("hello world")
print (name)
```

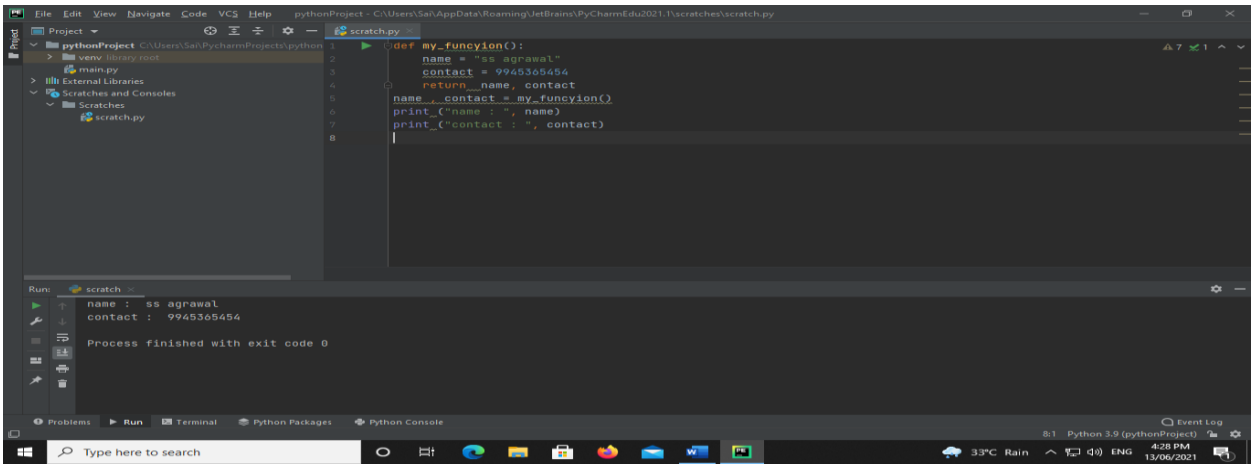
Output:-



Function with multiple return statement:-

```
def my_funcyion():
    name = "ss agrawal"
    contact = 9945365454
    return  name, contact
name , contact = my_funcyion()
print ("name : ", name)
print ("contact : ", contact)
```

Output:-

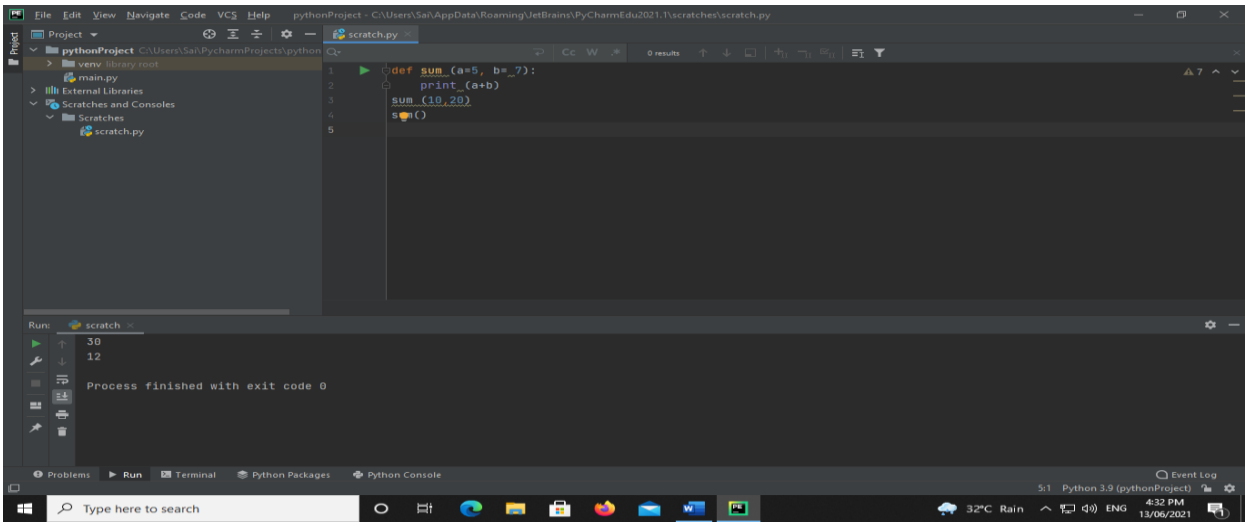


Python function argument:-

Code:

```
def sum (a=5, b= 7):
    print (a+b)
sum (10,20)
sum()
```

Output:-



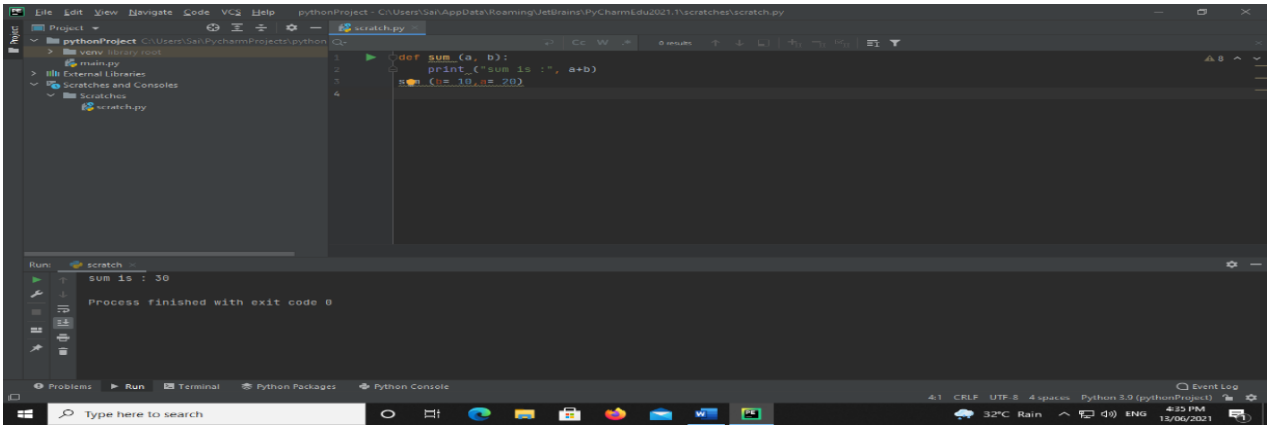
Python keyword arguments:

- In function you may need more argument to process function then you mentioned in the definition

Code:

```
def sum (a, b):
    print ("sum is :", a+b)
sum (b= 10,a= 20)
```

Output:-

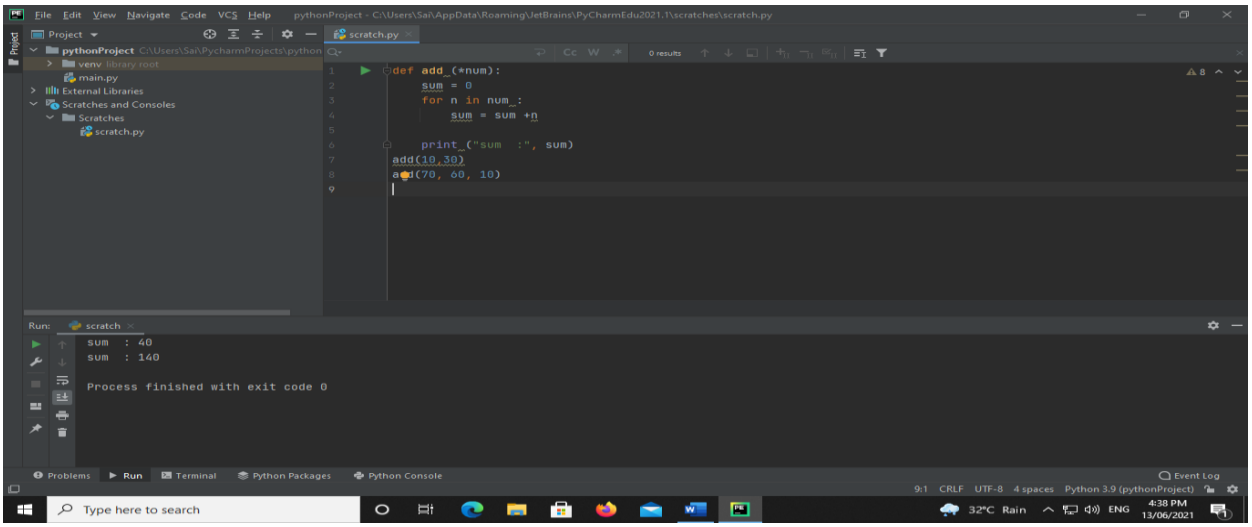


Non keyword arguments:-

Code:

```
def add (*num):
    sum = 0
    for n in num :
        sum = sum +n

    print ("sum  :", sum)
add(10,30)
add(70, 60, 10)
```

Output:-

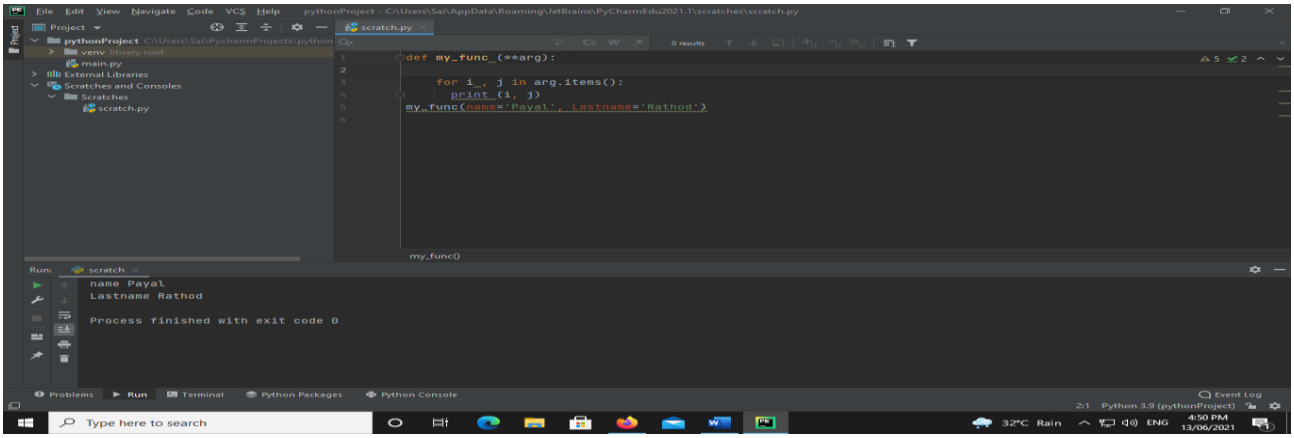


Keyword arguments:-

Code:

```
def my_func (**arg):

    for i , j in arg.items():
      print (i, j)
my_func(name='Payal', Lastname='Rathod')
```

Output:-



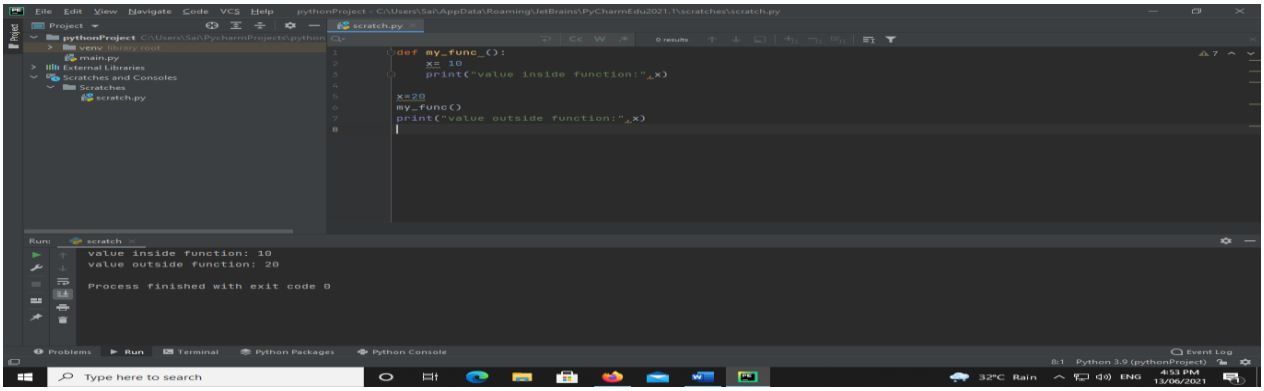
Scope of variable:

- **There are two Scope of variable in Python**
- **Global**
- **Local**

Code

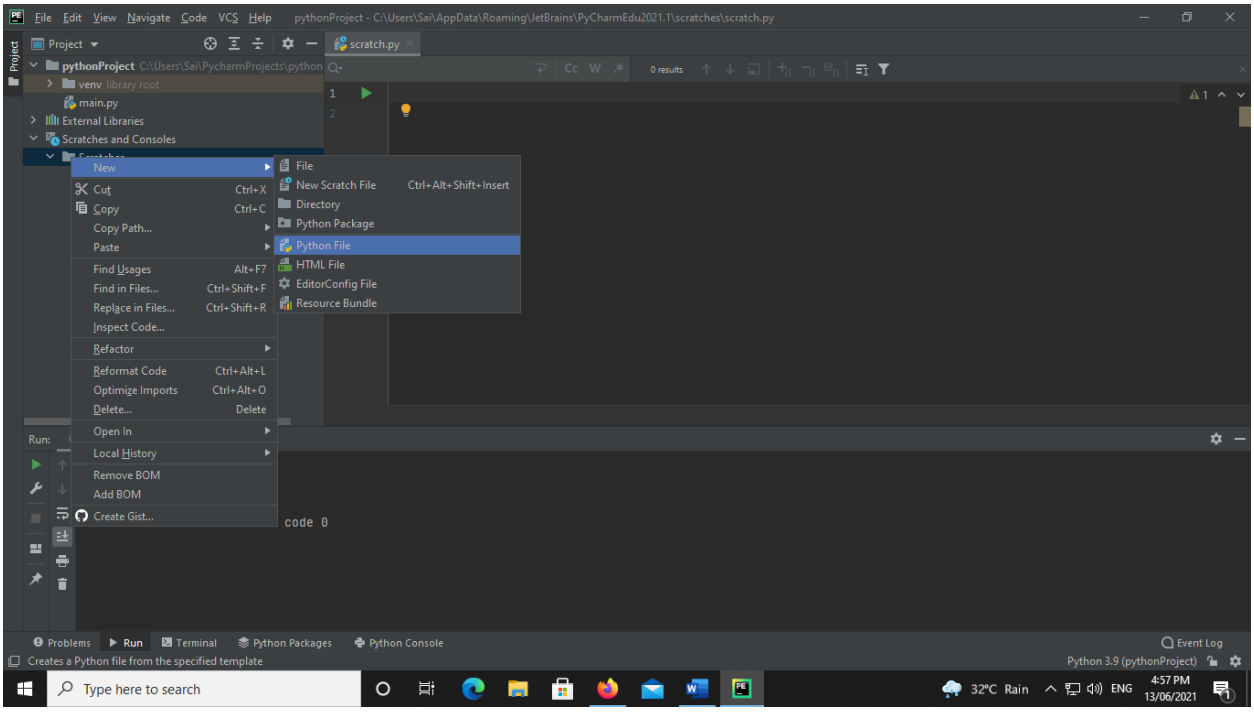
```
def my_func ():
    x= 10
    print("value inside function:",x)

x=20
my_func()
print("value outside function:",x)
```

Output:-

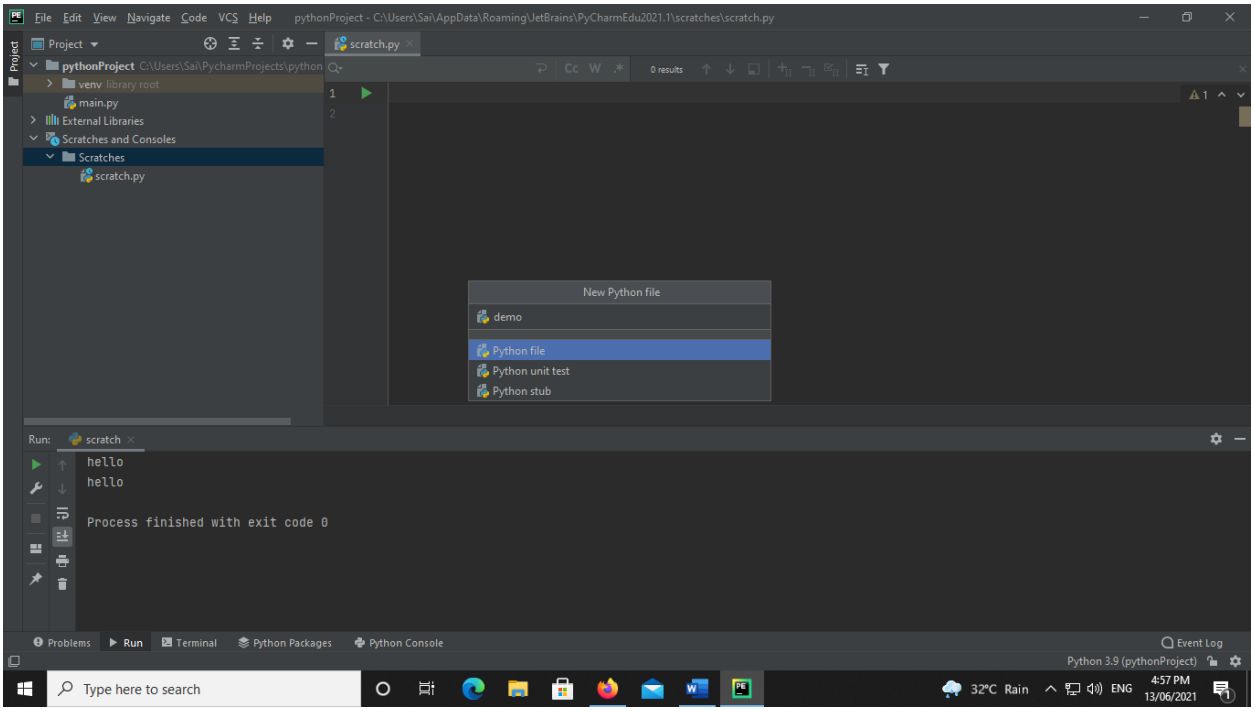


The Import Statement:-

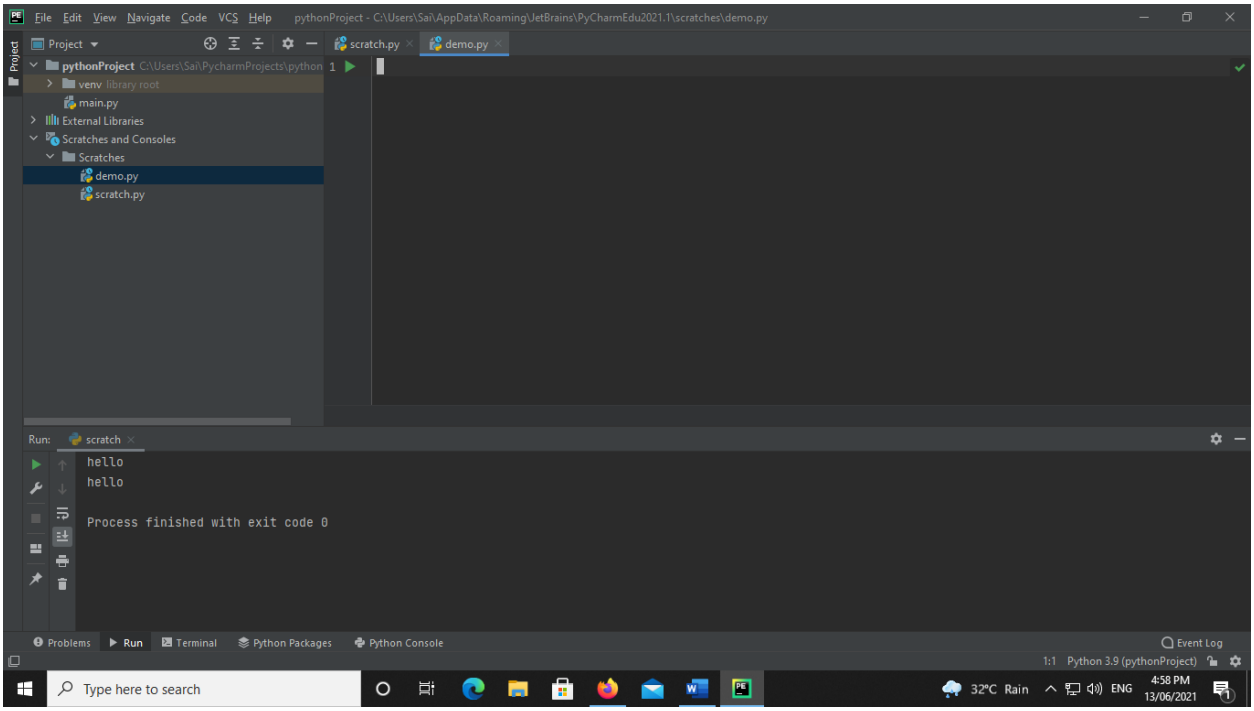
Create a new file by right clicking on Scratches folder:_



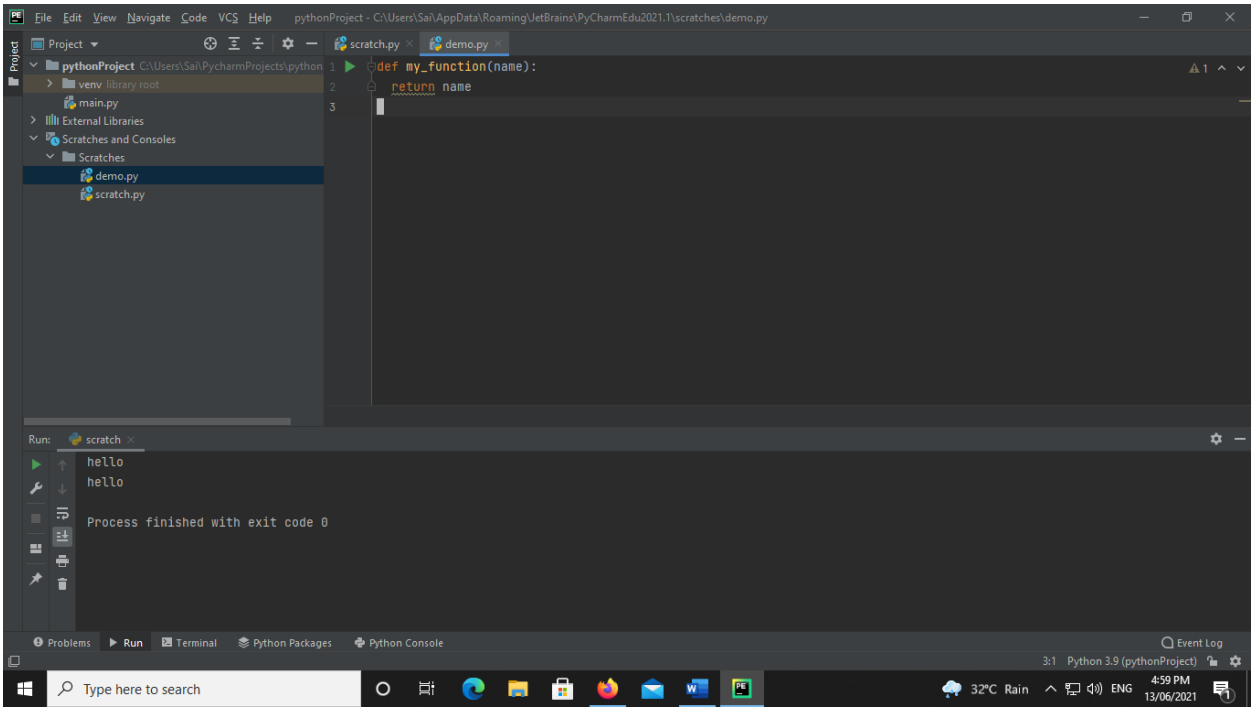
Give file name



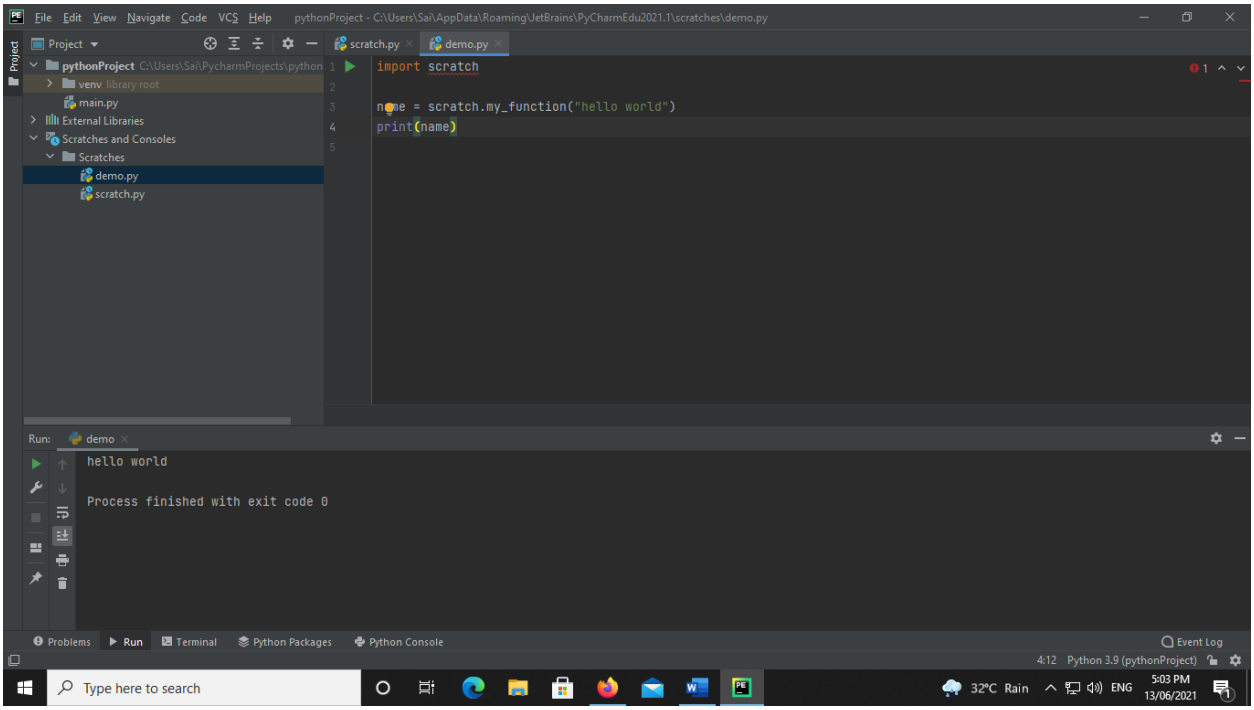
File is created



Scratch.py


File Edit View Navigate Code VCS Help
pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\demo.py
Project
pythonProject C:\Users\Sai\PycharmProjects\python
venv library root
main.py
External Libraries
Scratches and Consoles
Scratches
demo.py
scratch.py
scratch.py
demo.py
def my_function(name):
return name
Run: scratch
hello
hello
Process finished with exit code 0
Problems
Run
Terminal
Python Packages
Python Console
Event Log
3:1 Python 3.9 (pythonProject)
Type here to search
32°C Rain
ENG
4:59 PM
13/06/2021

Demo.py


File Edit View Navigate Code VCS Help
pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\demo.py
Project
pythonProject C:\Users\Sai\PycharmProjects\python
venv library root
main.py
External Libraries
Scratches and Consoles
Scratches
demo.py
scratch.py
scratch.py
demo.py
import scratch
name = scratch.my_function("hello world")
print(name)
Run: demo
hello world
Process finished with exit code 0
Problems
Run
Terminal
Python Packages
Python Console
Event Log
4:12 Python 3.9 (pythonProject)
Type here to search
32°C Rain
ENG
5:03 PM
13/06/2021

Operators in Python:

- Arithmetic Operators
- Comparison Operators
- Logical Operators
- Assignment Operators
- Membership Operators
- Identity Operators

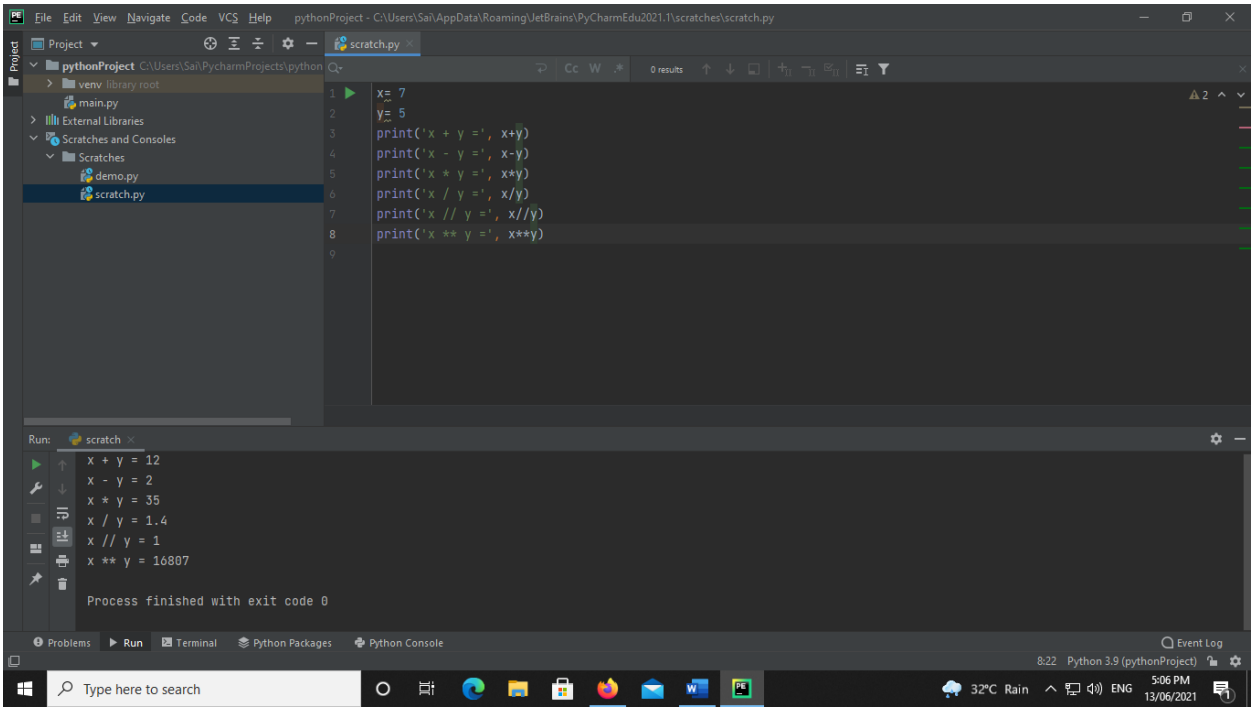
- **Arithmetic Operators**

Operator	Meaning
+	Add two operands or unary plus
-	Subtract right operand from the left or unary minus
*	Multiply two operands
/	Divide left operand by the right one (always results into float)
%	Modulus - remainder of the division of left operand by the right
//	Floor division - division that results into whole number adjusted to the left in the number line
**	Exponent - left operand raised to the power of right(x to the power y)

Code:

```
x= 7
y= 5
print('x + y =', x+y)
print('x - y =', x-y)
print('x * y =', x*y)
print('x / y =', x/y)
print('x // y =', x//y)
print('x ** y =', x**y)
```

Output:-



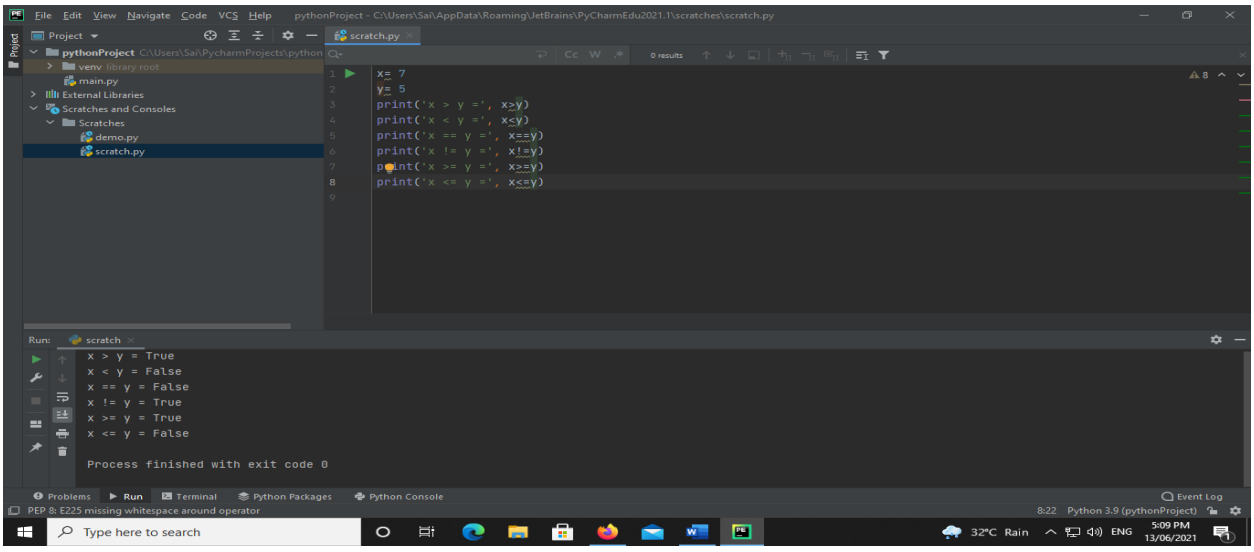
- **Comparison Operators**

Operator	Meaning	Example
>	Greater than	x > y
<	Less than	x < y
==	Equal to	x == y
!=	Not equal to	x != y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Code:-

```
x= 7
y= 5
print('x > y =', x>y)
print('x < y =', x<y)
print('x == y =', x==y)
print('x != y =', x!=y)
print('x >= y =', x>=y)
print('x <= y =', x<=y)
```

Output:-



- **Logical Operators**

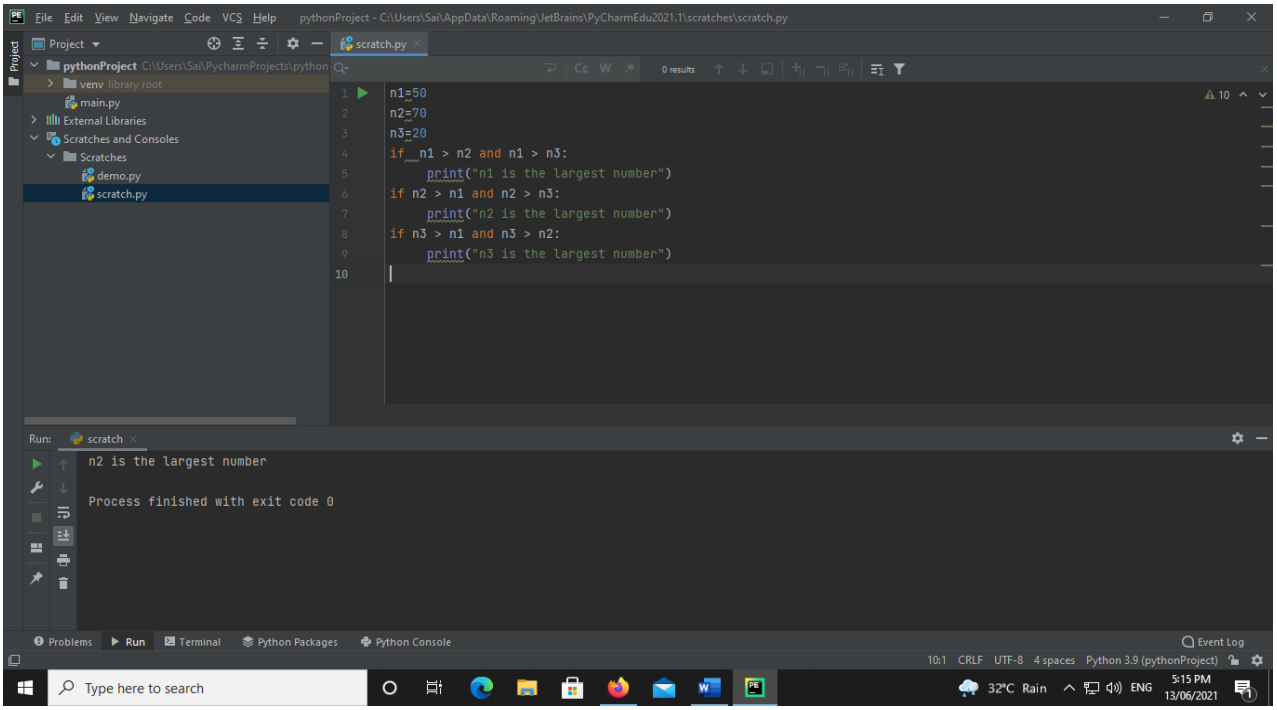
Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

And:-

Code

```
n1=50
n2=70
n3=20
if  n1 > n2 and n1 > n3:
     print("n1 is the largest number")
if n2 > n1 and n2 > n3:
     print("n2 is the largest number")
if n3 > n1 and n3 > n2:
     print("n3 is the largest number")
```

Output:-

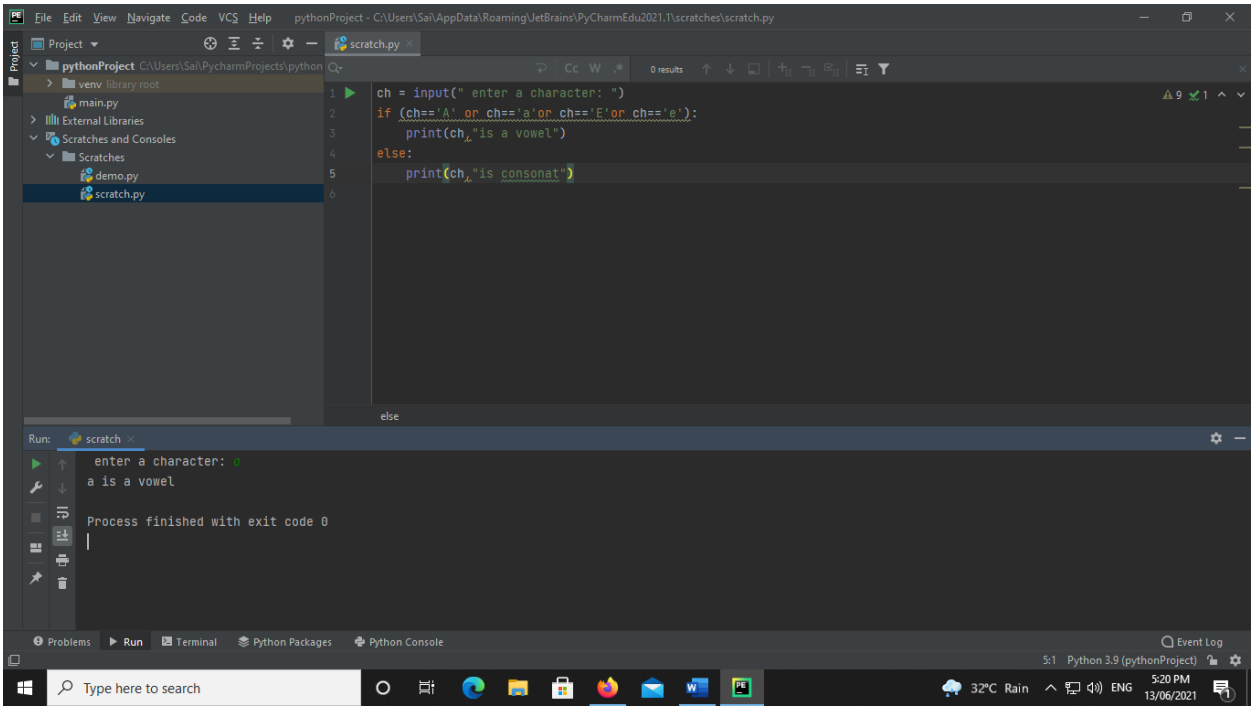


Or:-

Code:

```
ch = input(" enter a character: ")
if (ch=='A' or ch=='a'or ch=='E'or ch=='e'):
    print(ch,"is a vowel")
else:
    print(ch,"is consonat")
```

Output:-



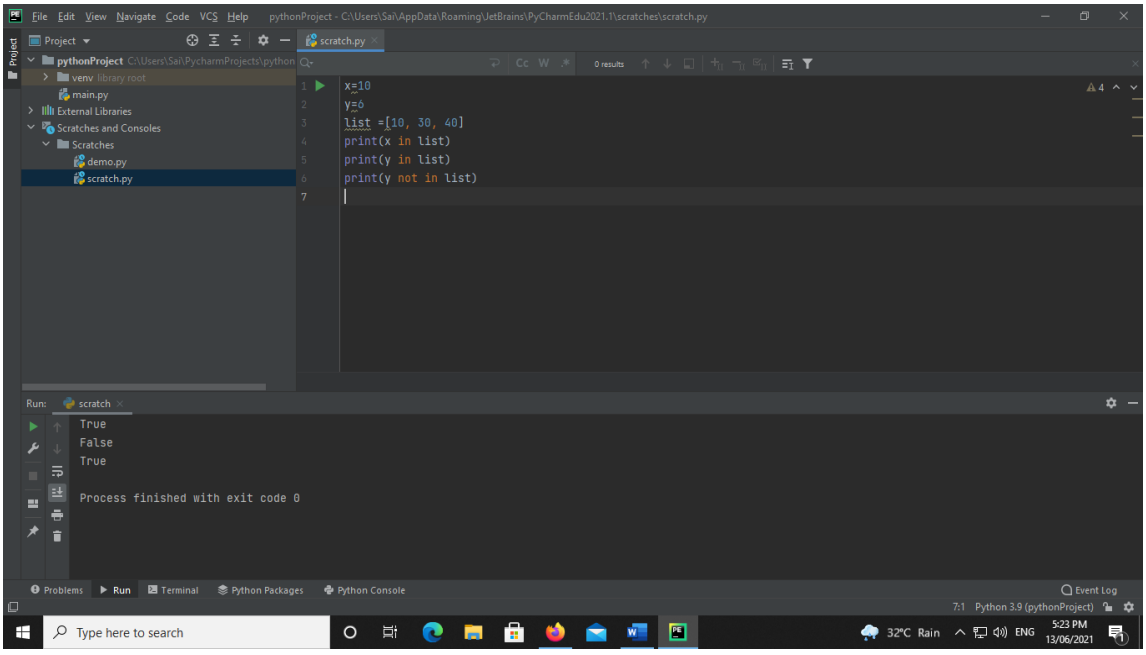
- **Membership Operators**

Operator	Meaning
in	True if value/variable is found in the sequence
not in	True if value/variable is not found in the sequence

Code

```
x=10
y=6
list =[10, 30, 40]
print(x in list)
print(y in list)
print(y not in list)
```

Output:-



Task:-5

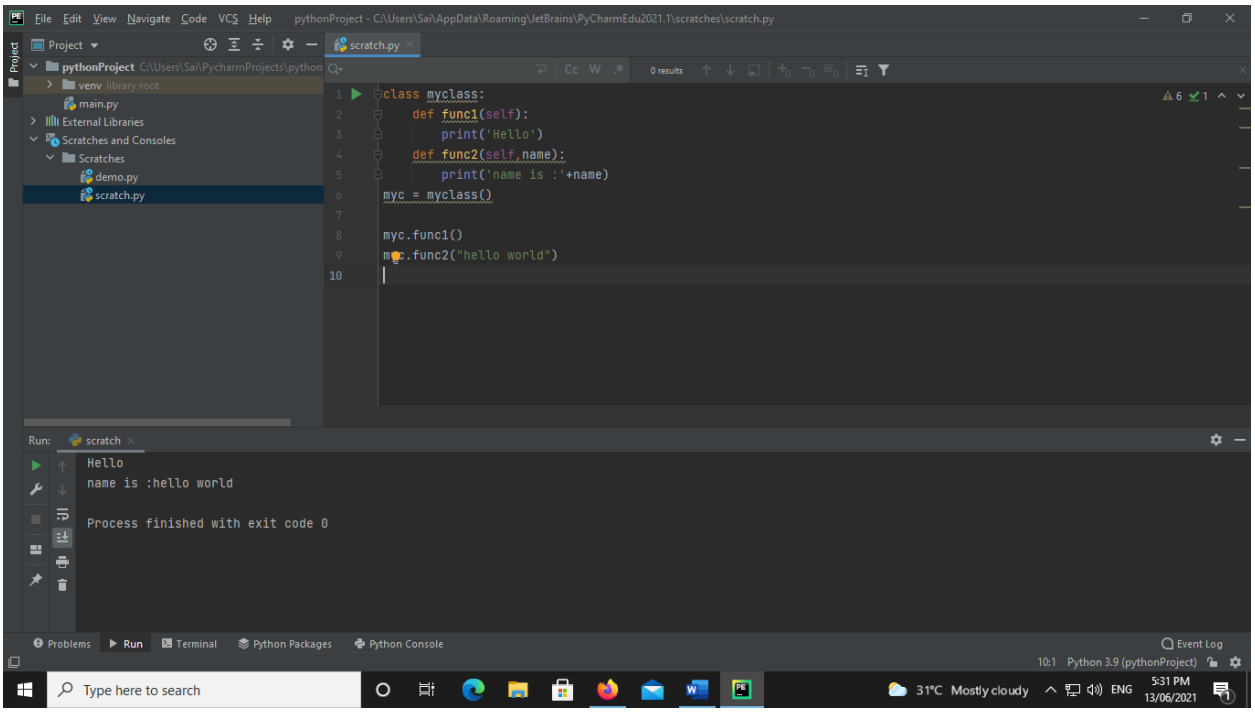
Class:

- A class is blueprint for the object.
- **Syntax:**

Class Myclass:

```
 class myclass:
    def func1(self):
        print('Hello')
    def func2(self,name):
        print('name is :'+name)
myc = myclass()

myc.func1()
myc.func2("hello world")
```

Output:

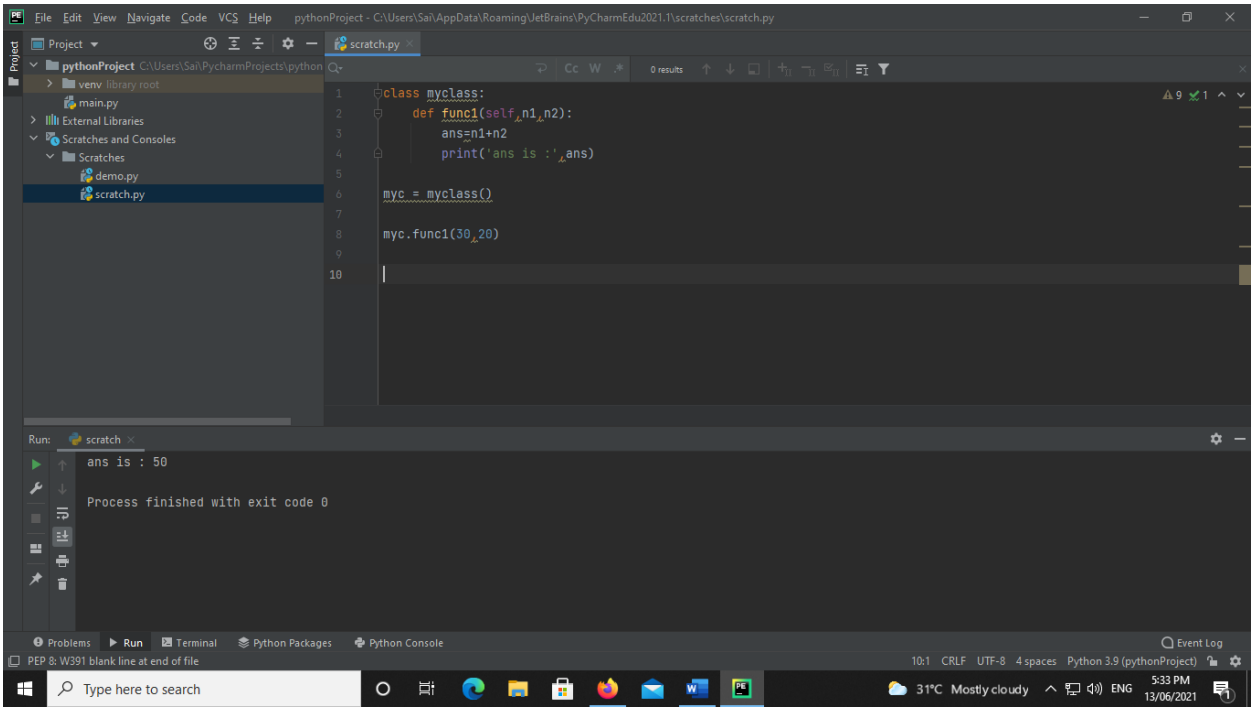


Sum of 2 number using class:-

Code:

```
class myclass:
    def func1(self,n1,n2):
        ans=n1+n2
        print('ans is :',ans)

myc = myclass()

myc.func1(30,20)
```

Output:-



Python Constructors:-

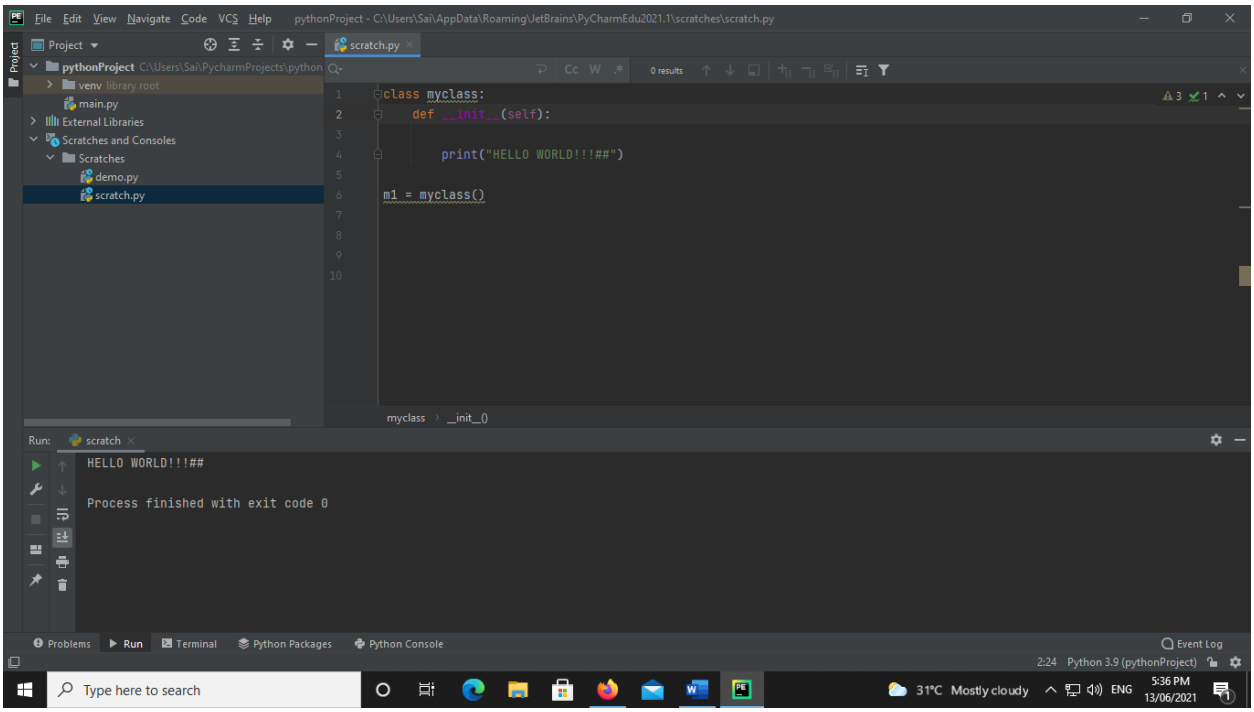
- A constructors is a class function the instantiates an object to predefined values.
- There are two type of Constructor.
- Default constructor
- Parameterized constructor

- **Default constructor**

Code:

```
class myclass:
    def __init__(self):

        print("HELLO WORLD!!!##")

m1 = myclass()
```

Output:-

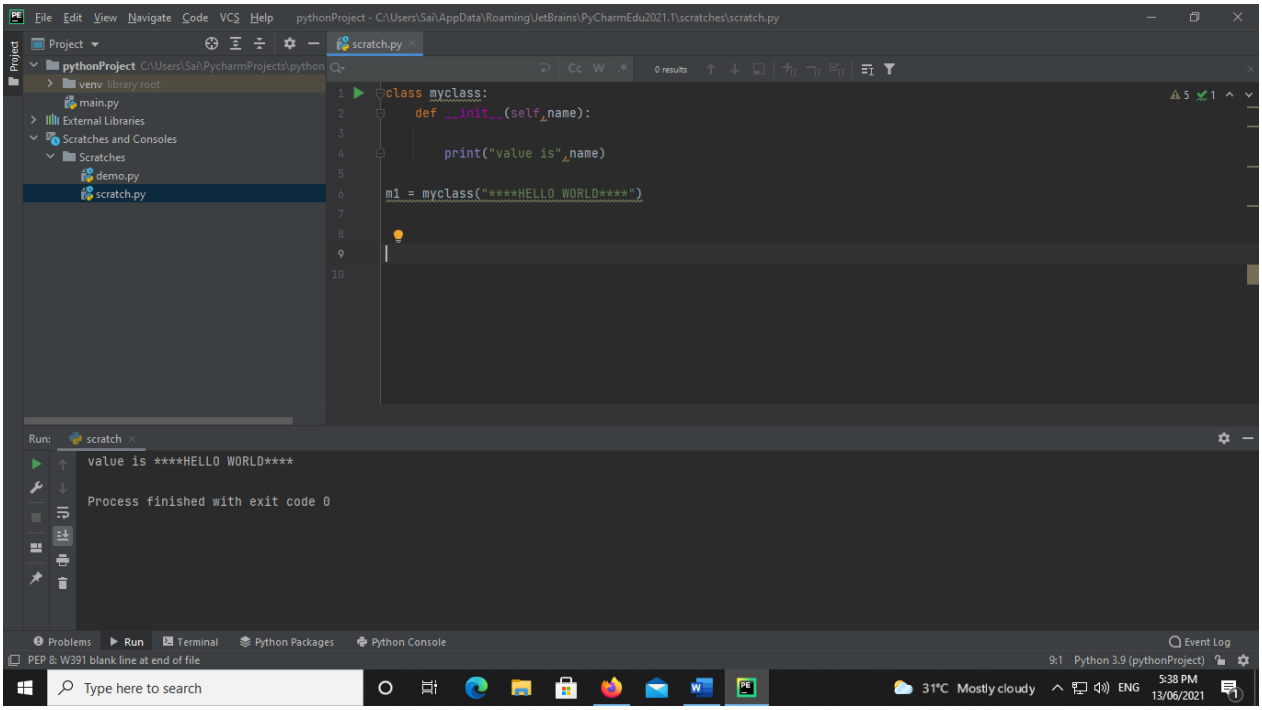


- **Parameterized constructor**

Code:

```
class myclass:
    def __init__(self,name):

        print("value is",name)

m1 = myclass("****HELLO WORLD****")
```

Output:-

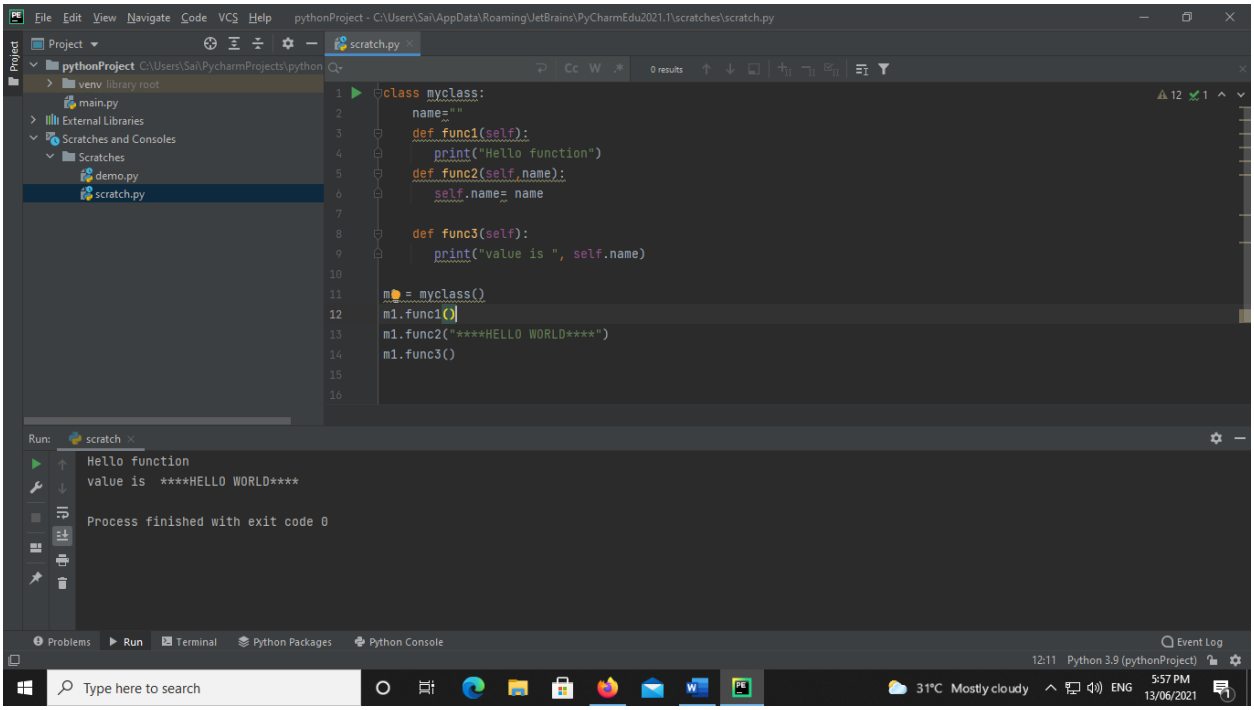


Assign string value to class variable using Method

Code:

```
class myclass:
    name=""
    def func1(self):
        print("Hello function")
    def func2(self,name):
        self.name= name

    def func3(self):
        print("value is ", self.name)

m1 = myclass()
m1.func1()
m1.func2("****HELLO WORLD****")
m1.func3()
```

Output:-

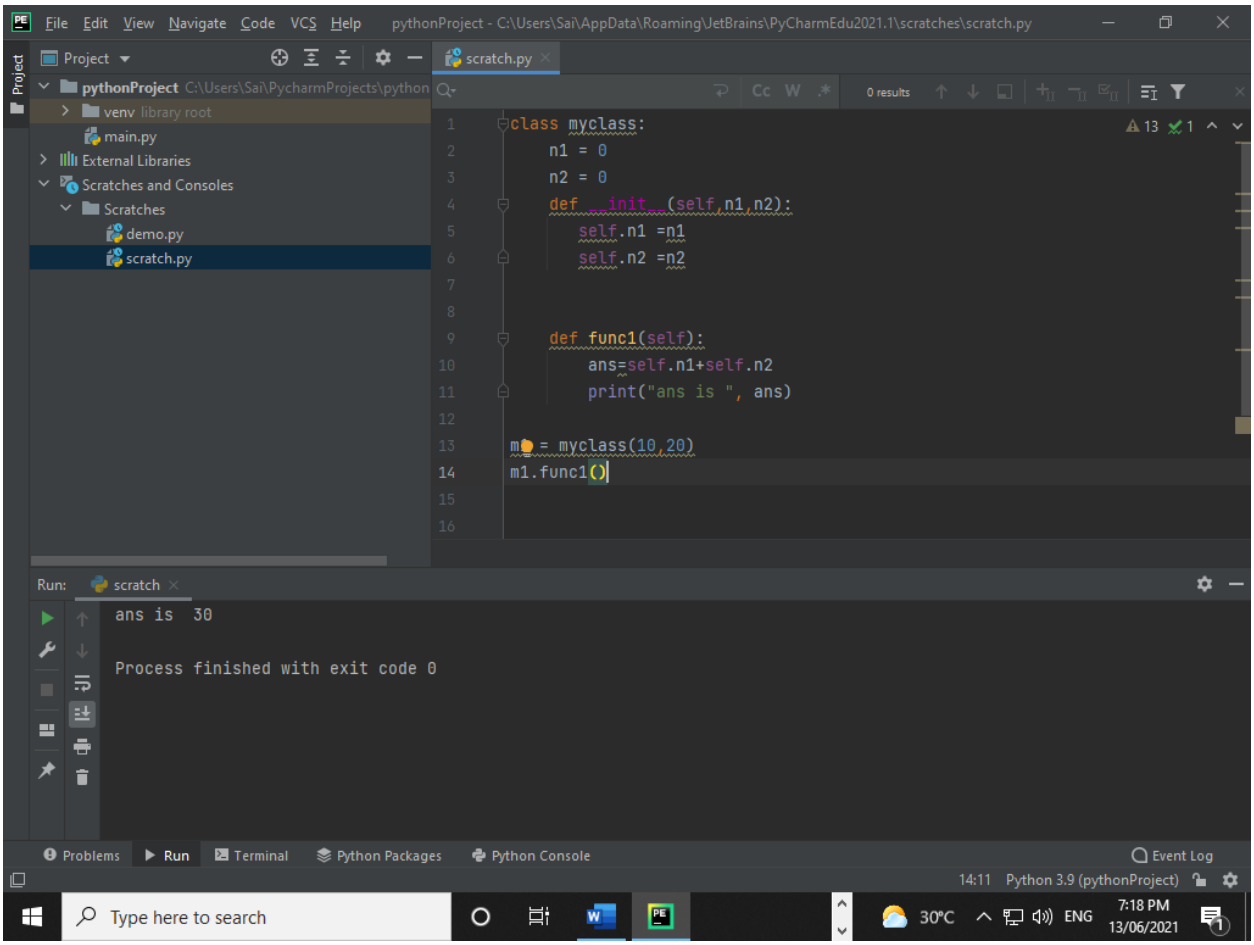


Assign string value to class variable using Constructor

Code:

```
class myclass:
    n1 = 0
    n2 = 0
    def __init__(self,n1,n2):
        self.n1 =n1
        self.n2 =n2


    def func1(self):
        ans=self.n1+self.n2
        print("ans is ", ans)

m1 = myclass(10,20)
m1.func1()
```

Output:-



Inheritance

Inheritance allows programmer to create a general class first then later extend it to more specialized class.

Type of Inheritance

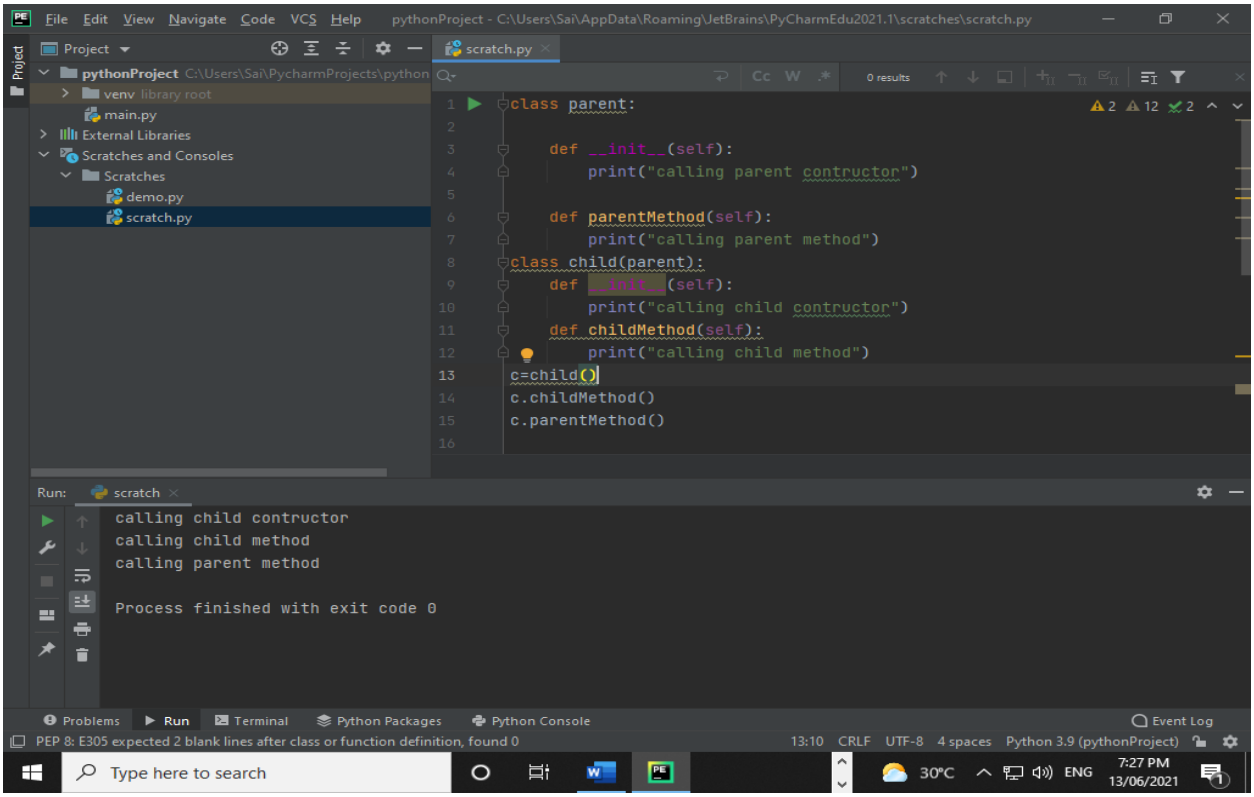
- Single-level Inheritance
- Multi-level Inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

- Single-level Inheritance

Code:

```
class parent:

    def __init__(self):
        print("calling parent contructor")

    def parentMethod(self):
        print("calling parent method")
class child(parent):
    def __init__(self):
        print("calling child contructor")
    def childMethod(self):
        print("calling child method")
c=child()
c.childMethod()
c.parentMethod()
```

Output:-

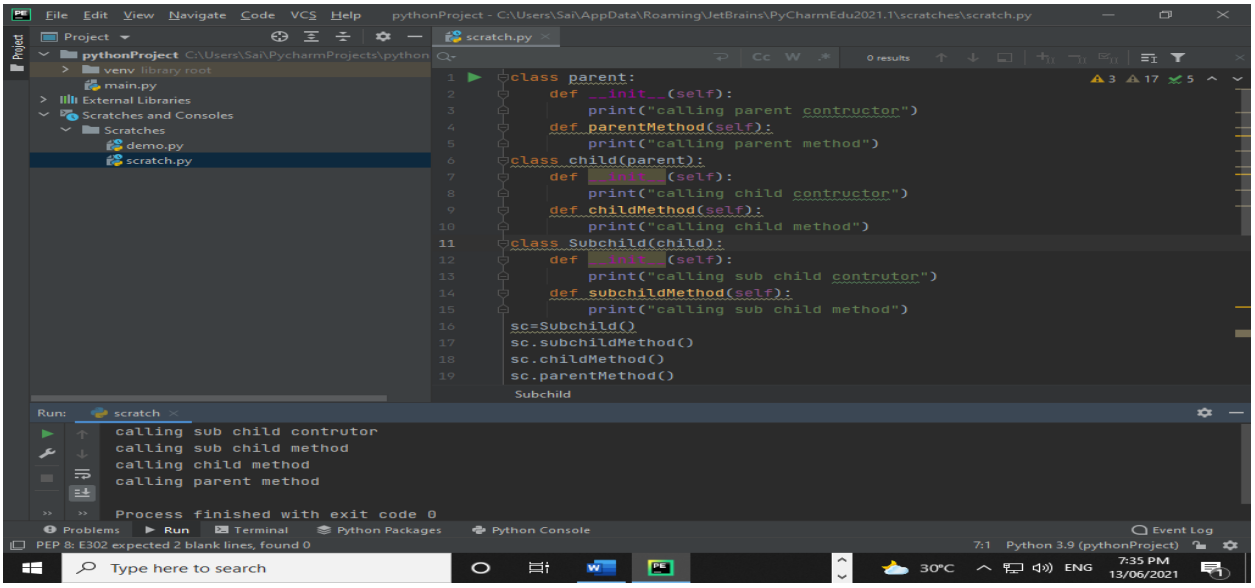

File Edit View Navigate Code VCS Help
pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py
Project
pythonProject C:\Users\Sai\PycharmProjects\python
venv library root
main.py
External Libraries
Scratches and Consoles
Scratches
demo.py
scratch.py
scratch.py
class parent:
def __init__(self):
print("calling parent contructor")
def parentMethod(self):
print("calling parent method")
class child(parent):
def __init__(self):
print("calling child contructor")
def childMethod(self):
print("calling child method")
c=child()
c.childMethod()
c.parentMethod()
Run: scratch
calling child contructor
calling child method
calling parent method
Process finished with exit code 0
Problems Run Terminal Python Packages Python Console Event Log
PEP 8: E305 expected 2 blank lines after class or function definition, found 0
13:10 CRLF UTF-8 4 spaces Python 3.9 (pythonProject)
Type here to search
30°C ENG 7:27 PM 13/06/2021

- Multi-level Inheritance

Code:

```
class parent:
    def __init__(self):
        print("calling parent contructor")
    def parentMethod(self):
        print("calling parent method")
class child(parent):
    def __init__(self):
        print("calling child contructor")
    def childMethod(self):
        print("calling child method")
class Subchild(child):
    def __init__(self):
        print("calling sub child contrutor")
    def subchildMethod(self):
        print("calling sub child method")
sc=Subchild()
sc.subchildMethod()
sc.childMethod()
sc.parentMethod()
```

Output:-

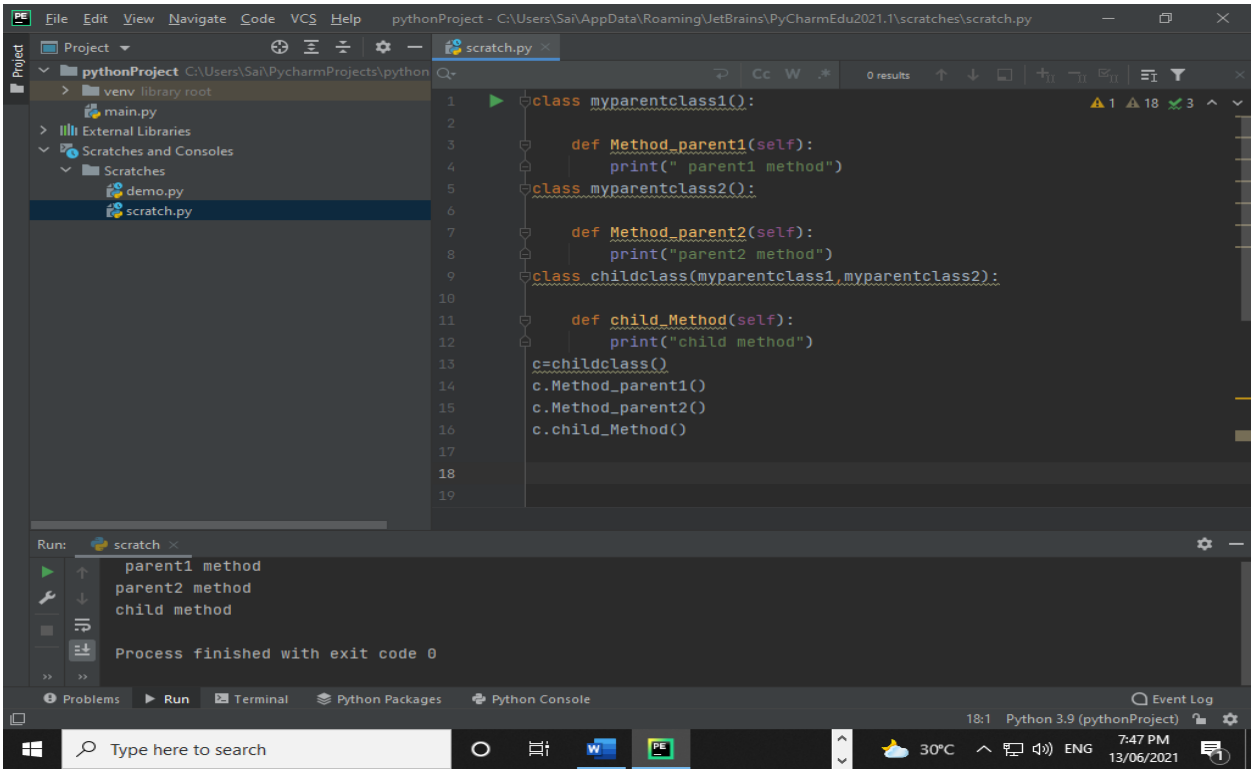


- Multiple Inheritance

Code:

```
class myparentclass1():

    def Method_parent1(self):
        print(" parent1 method")
class myparentclass2():

    def Method_parent2(self):
        print("parent2 method")
class childclass(myparentclass1,myparentclass2):

    def child_Method(self):
        print("child method")
c=childclass()
c.Method_parent1()
c.Method_parent2()
c.child_Method()
```

Output:

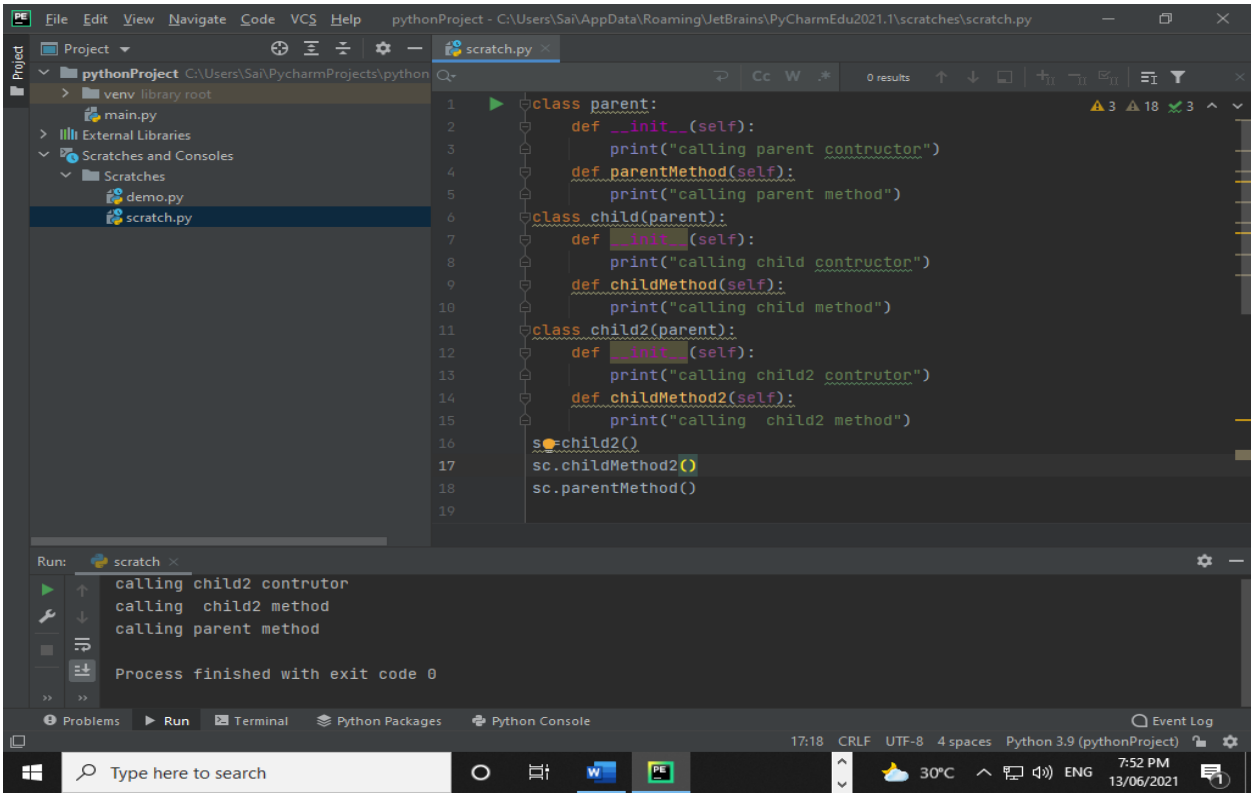


- Hierarchical Inheritance

Code:

```
class parent:
    def __init__(self):
        print("calling parent contructor")
    def parentMethod(self):
        print("calling parent method")
class child(parent):
    def __init__(self):
        print("calling child contructor")
    def childMethod(self):
        print("calling child method")
class child2(parent):
    def __init__(self):
        print("calling child2 contrutor")
    def childMethod2(self):
        print("calling  child2 method")
sc=child2()
sc.childMethod2()
sc.parentMethod()
```

Output:-

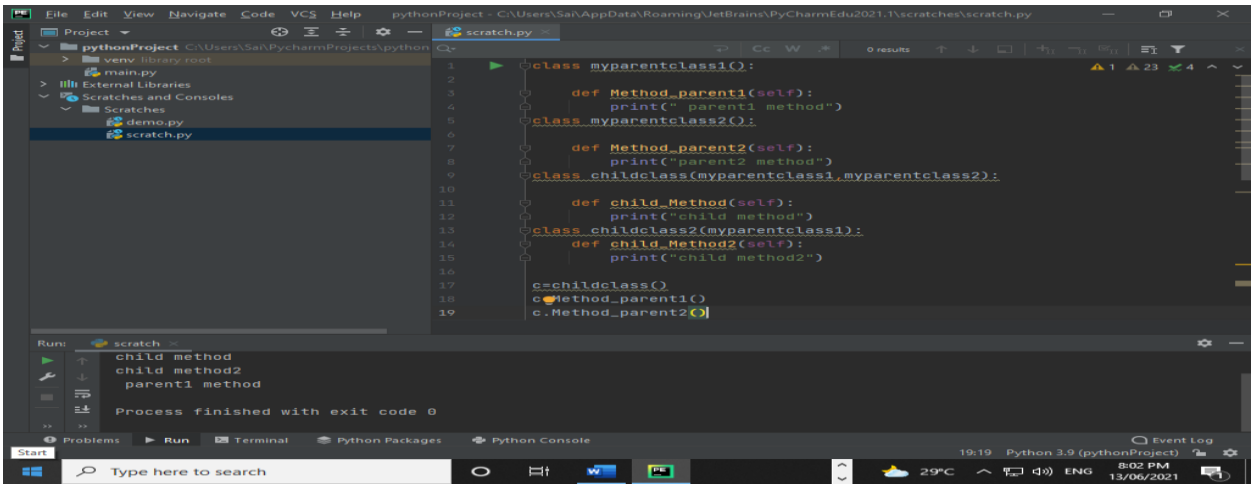


- Hybrid Inheritance

Code:

```
class myparentclass1():

    def Method_parent1(self):
        print(" parent1 method")
class myparentclass2():

    def Method_parent2(self):
        print("parent2 method")
class childclass(myparentclass1,myparentclass2):

    def child_Method(self):
        print("child method")
class childclass2(myparentclass1):
    def child_Method2(self):
        print("child method2")

c=childclass()
c.Method_parent1()
c.Method_parent2()
c.child_Method()

c=childclass2()
c.child_Method2()
c.Method_parent1()
```

Output:-

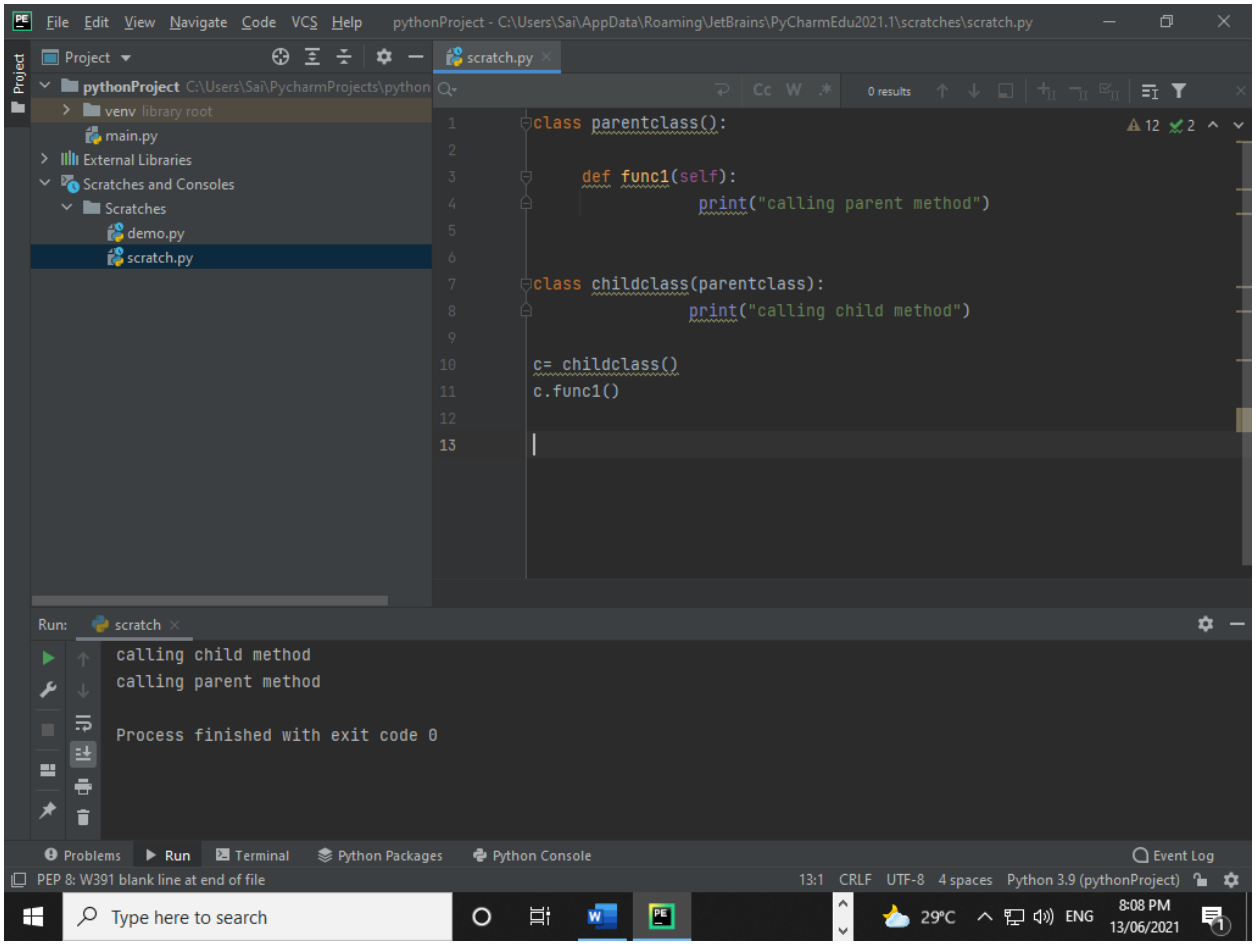


Polymorphism

- Polymorphism is an ability (in OOP) to use common interface for multiple from(data type).
- Type of Polymorphism:-
- Overloading Method
- Overriding Method

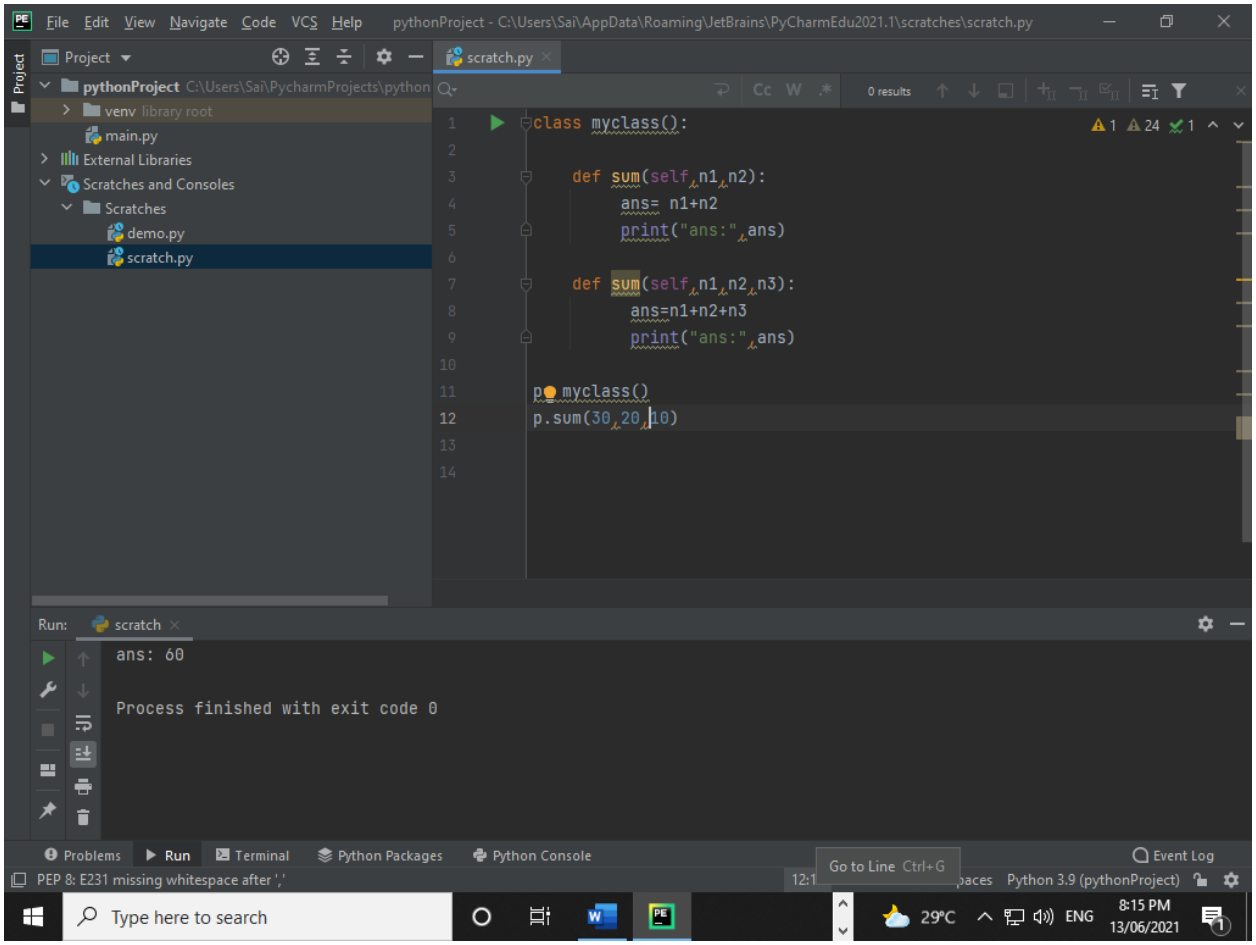
- Overloading Method

Code:



- Overriding Method

Code:



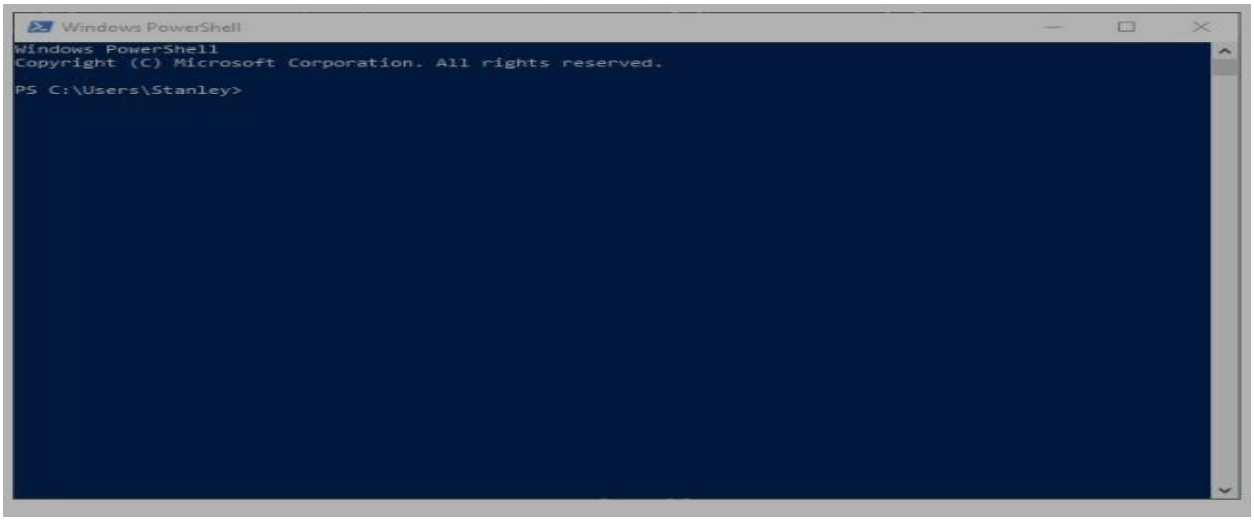
Task:6

Django:-

- Django is free and Opensource web application Framework , written Python
- Django framework follows the dry(don't repeat yourself)

Django install :-

Step 1- Open Powershell



Step 2- verify Python installation

```
PS C:\Users\Stanley> python -V
Python 3.7.4
```

Step 3- Upgrade pip

```
> python -m pip install --upgrade pip
```

Step 4-Create project

```
> mkdir django_project
```

```
> cd django_project
```

```
PS C:\Users\Username\django_project>
```

Step 5-create virtual environment

```
> python -m venv venv
```

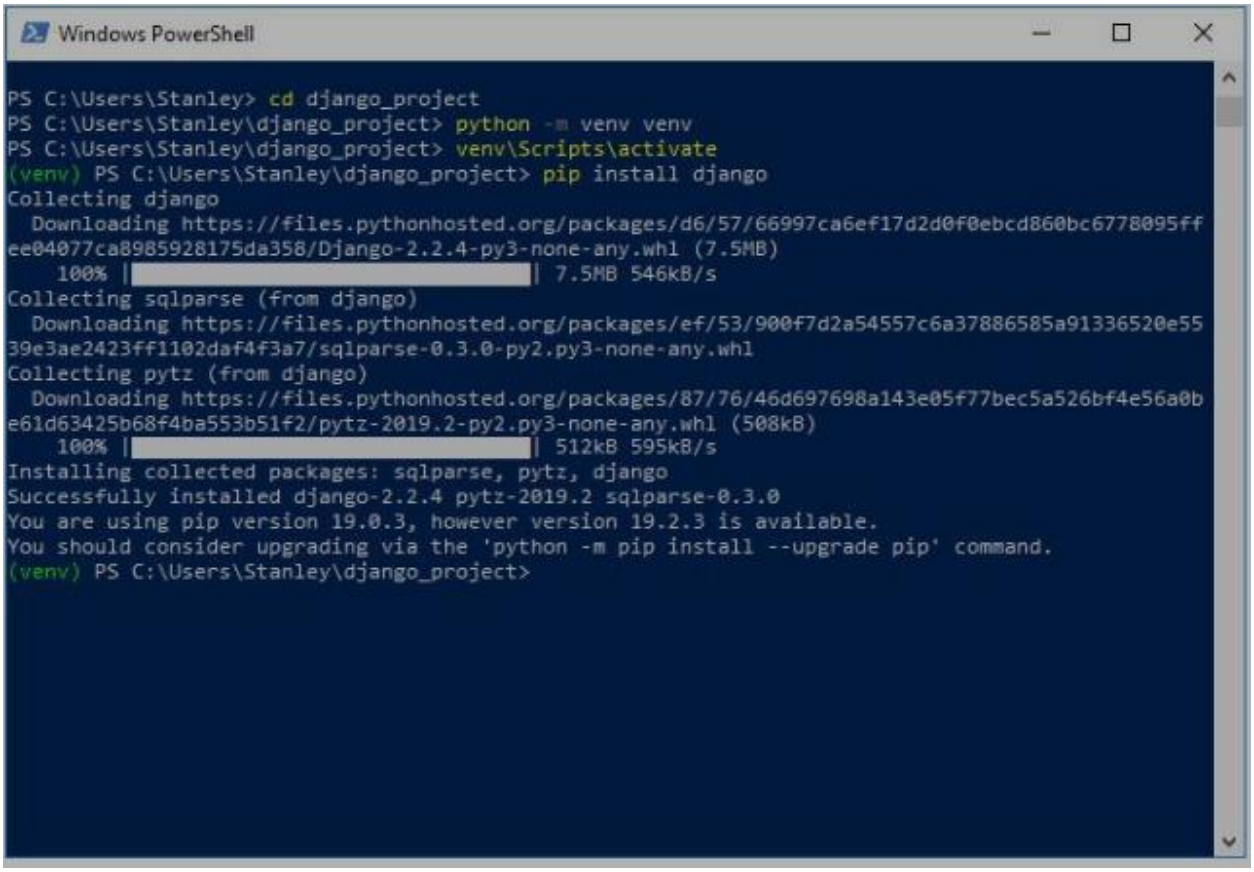
Step 6-Activate virtual environment

```
> venv\Scripts\activate
```

```
(venv) PS C:\Users\Stanley\django_project>
```

Step 7-install Django

```
(venv)> pip install django
```

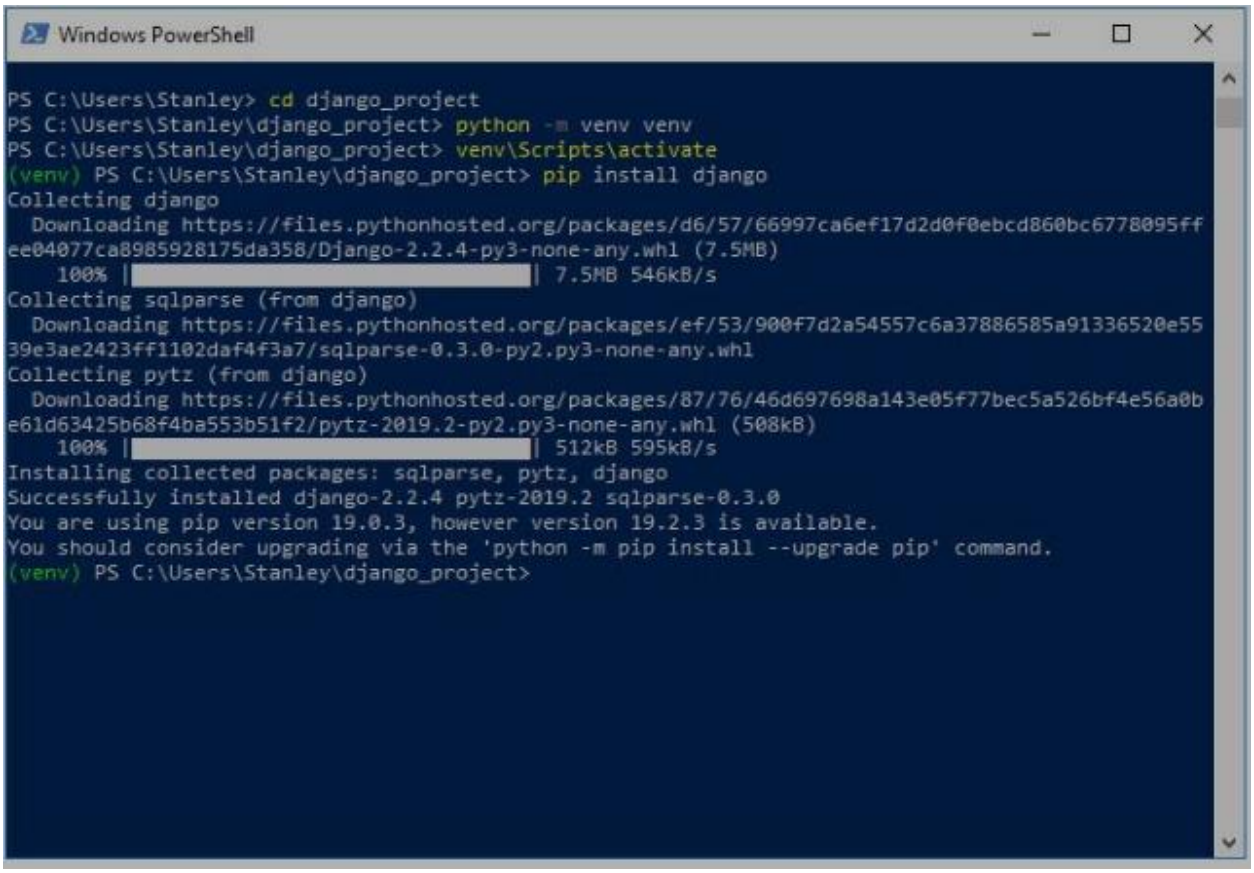


Step 8-Start new Project

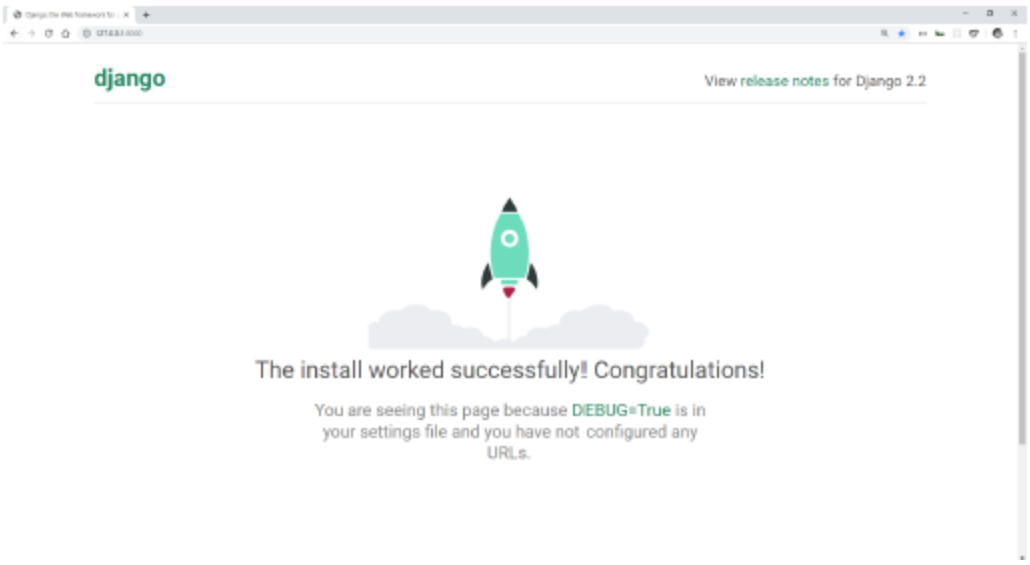
```
(venv)> django-admin startproject testsite
```

```
(venv)> cd testsite
```

Step 9-Run the server



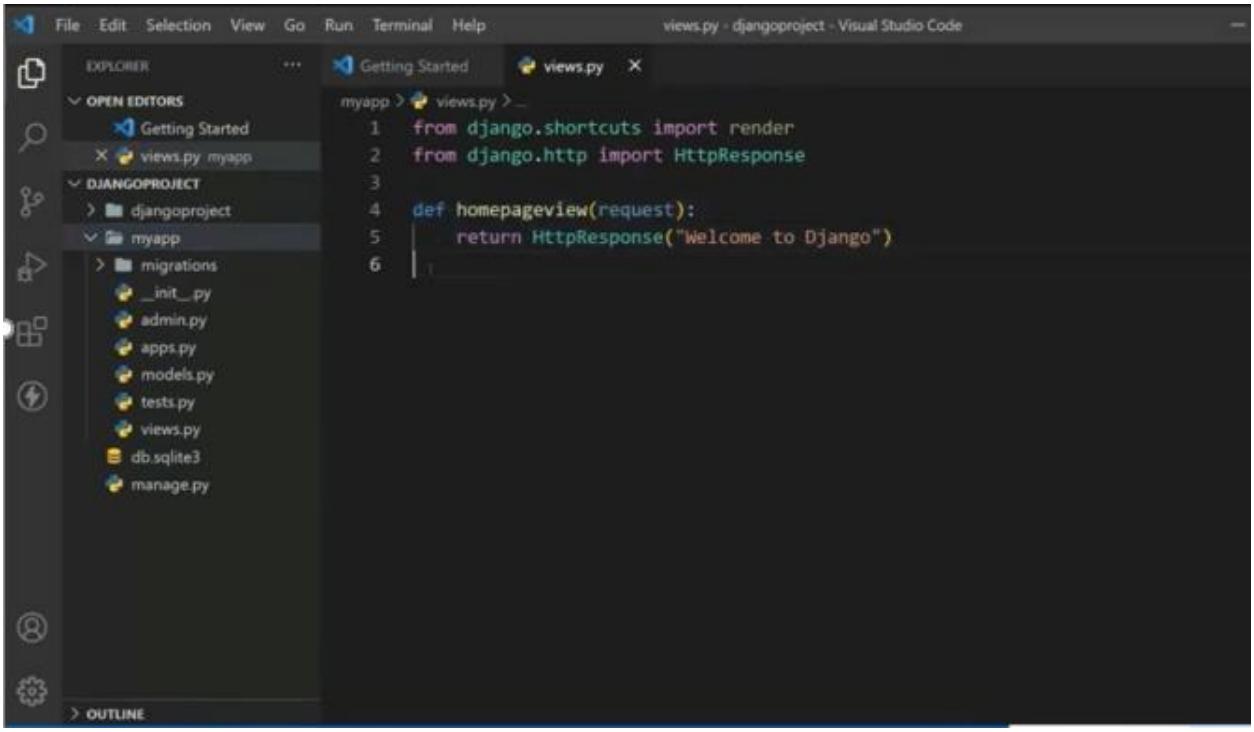
Welcome Screen


django
View release notes for Django 2.2
The install worked successfully! Congratulations!
You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

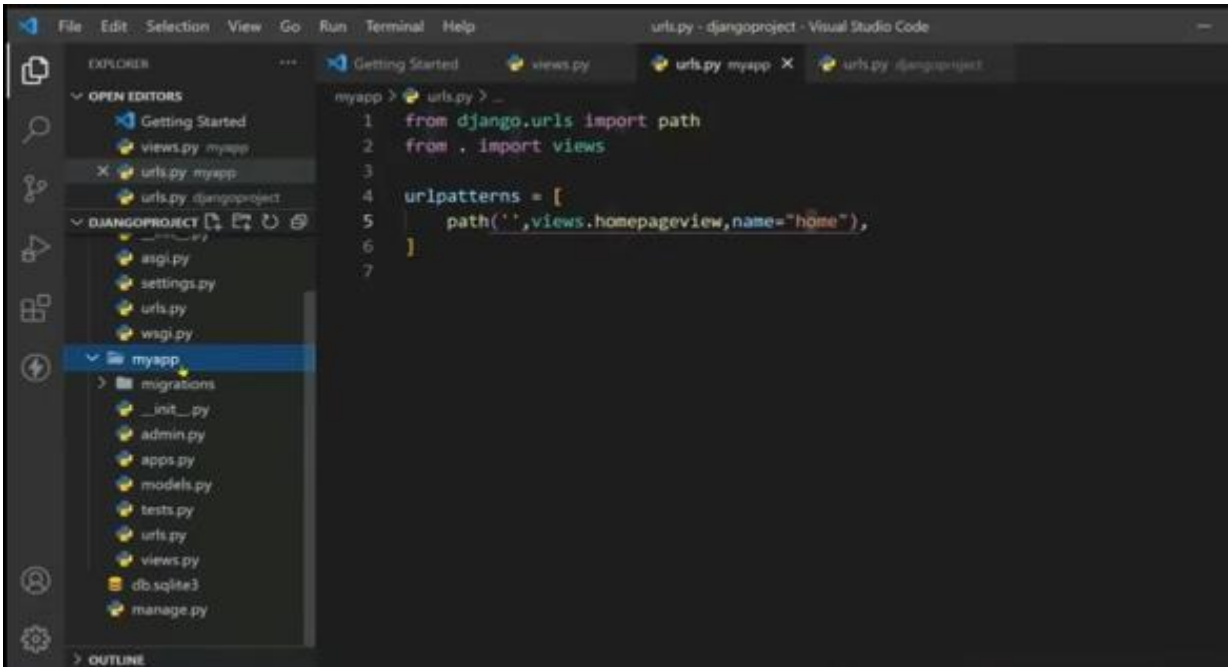
Task:7

Print Welcome to Django.

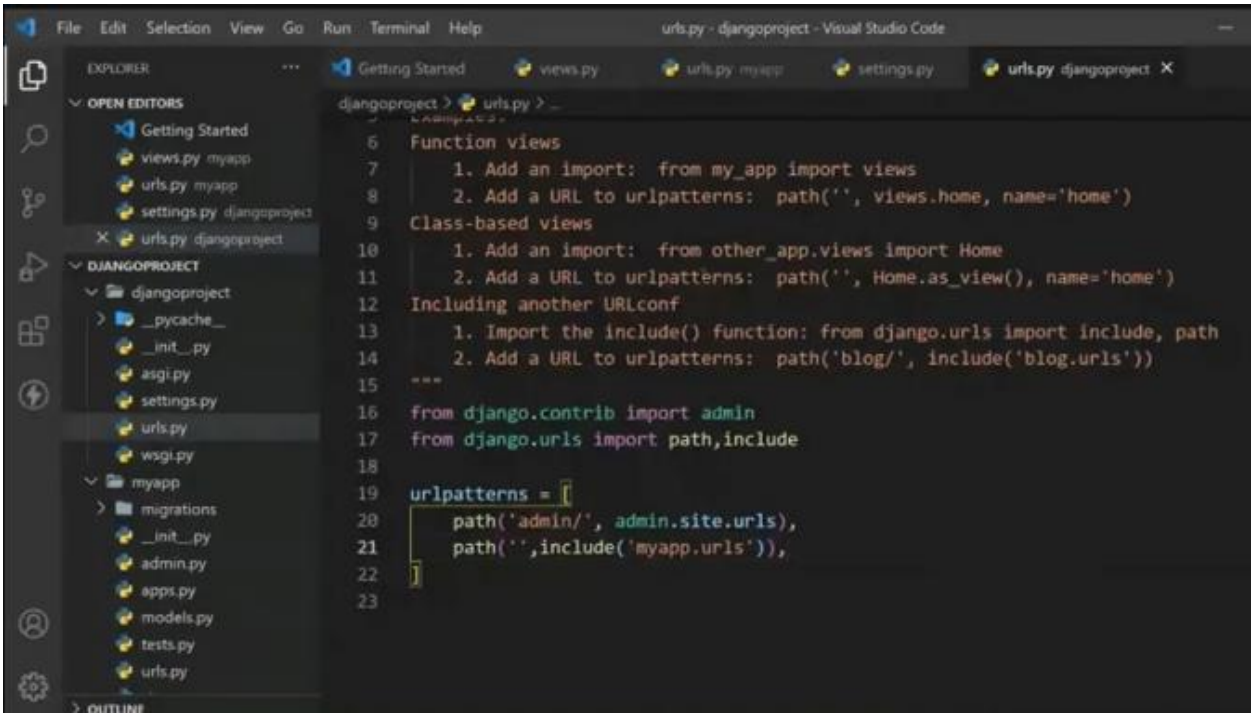
views.py



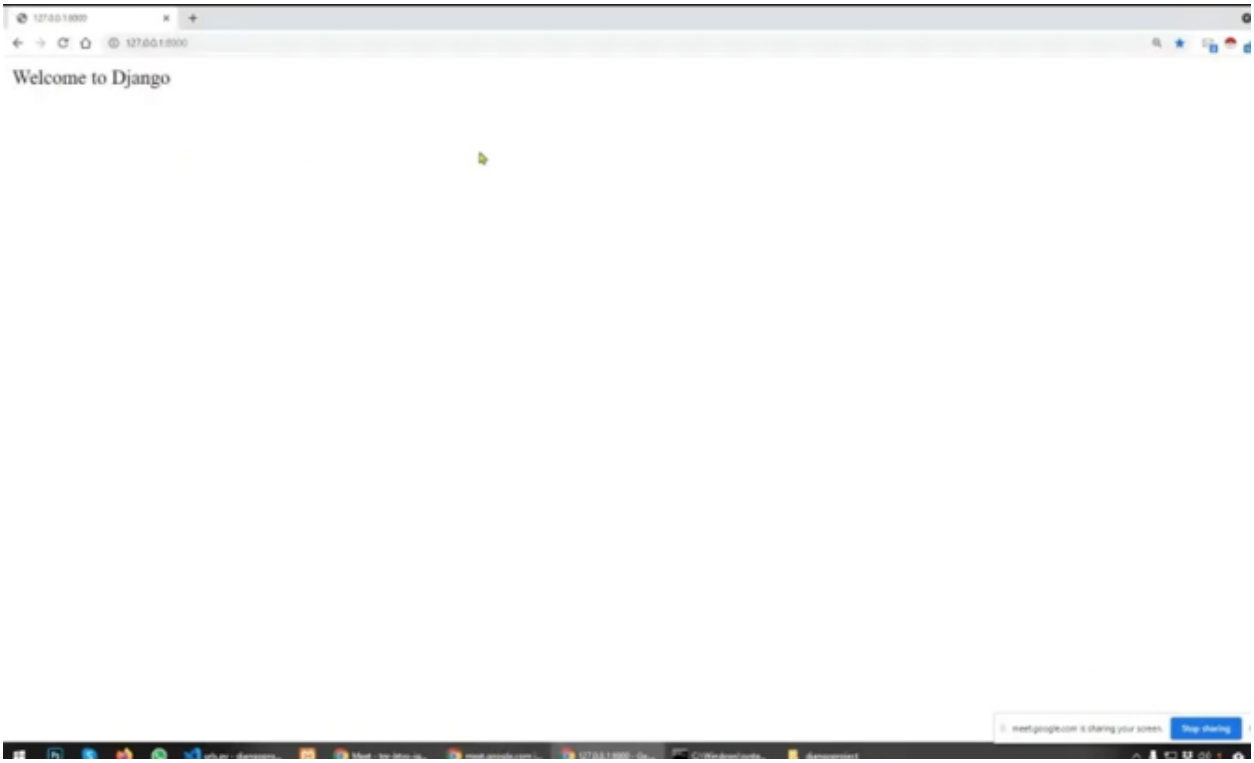
Urls.py my app



Url.py

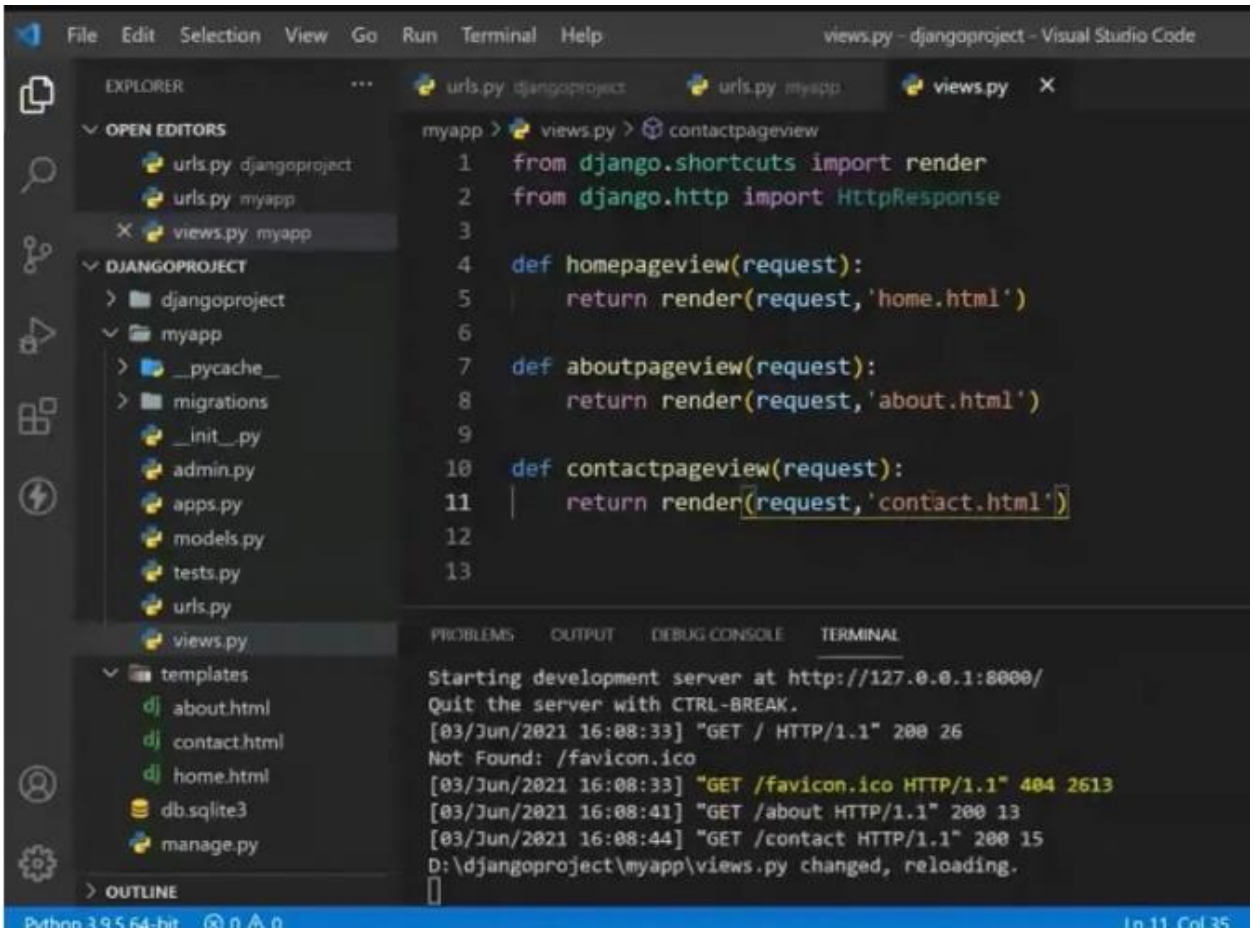


Output:

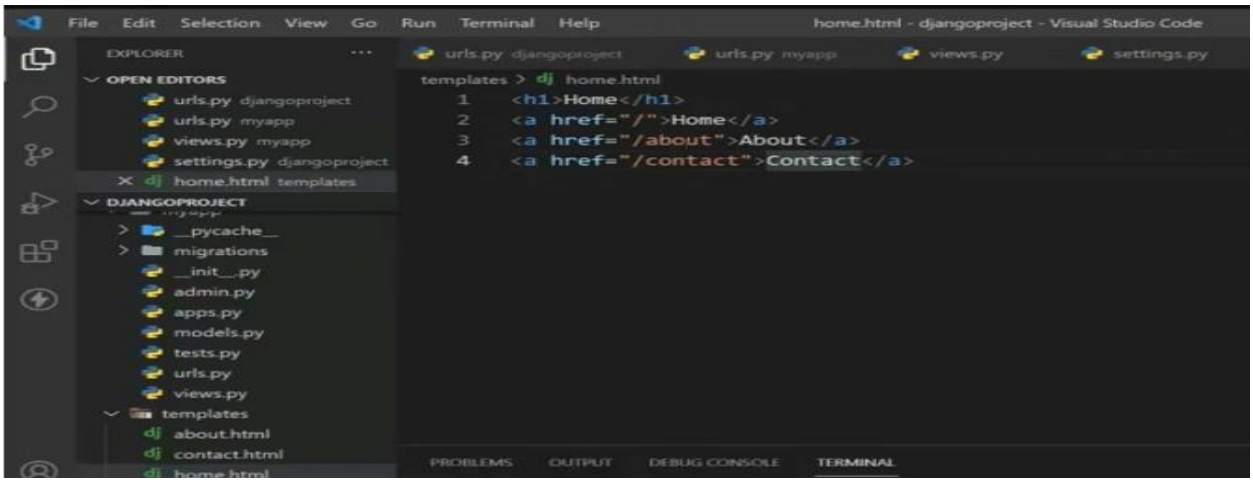

Welcome to Django

Task:8

In Website How to create pages in Django



Navigation bar

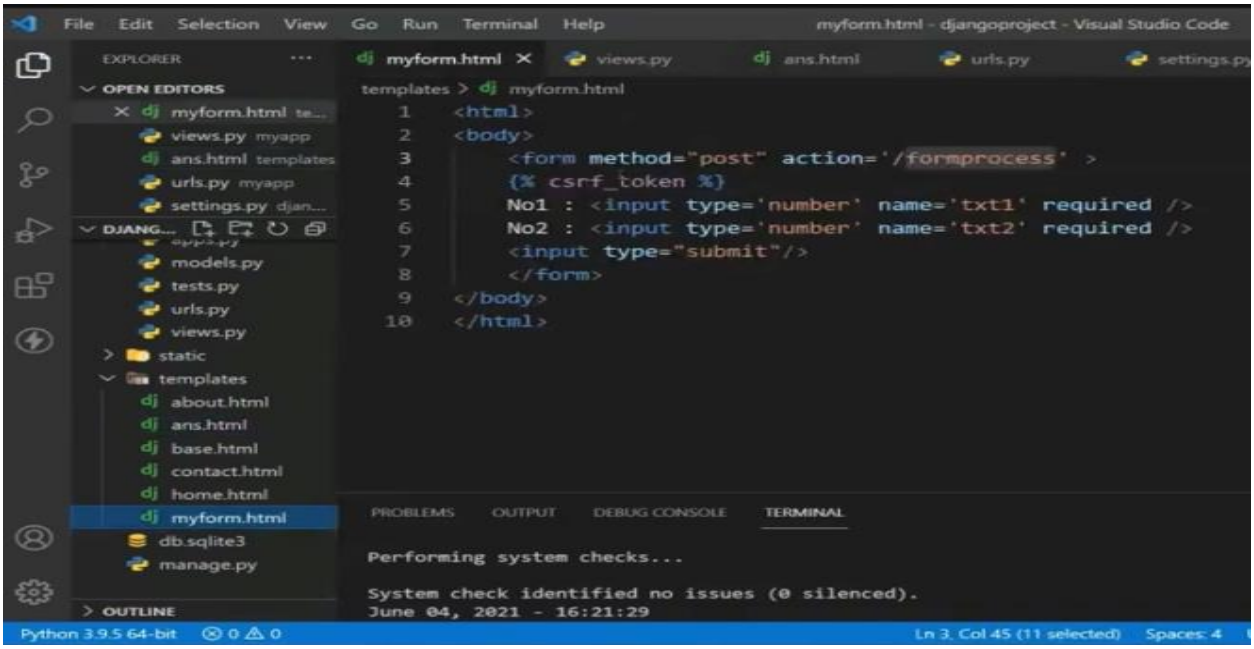


Output:-

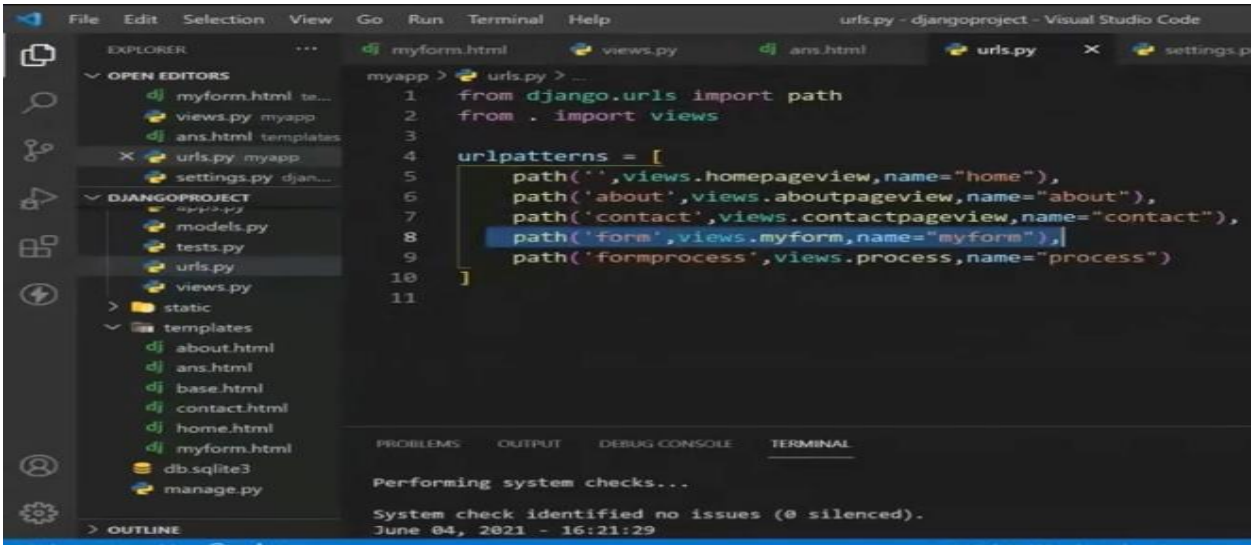
Task:9

How to Take value from user using Get and post method.

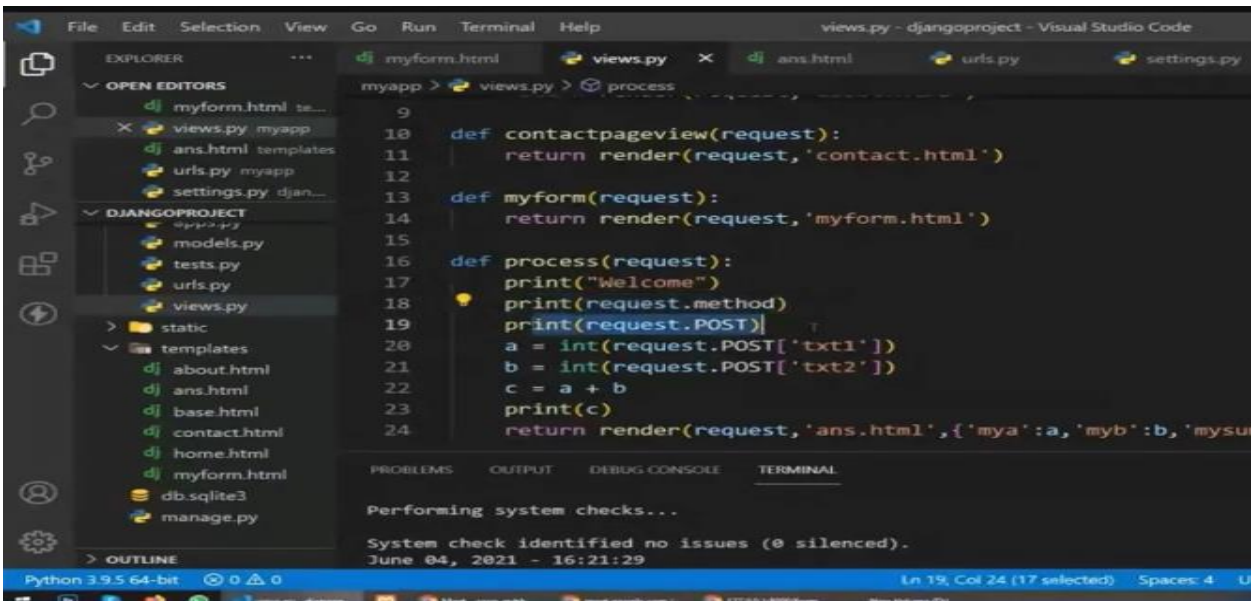
Myform.html



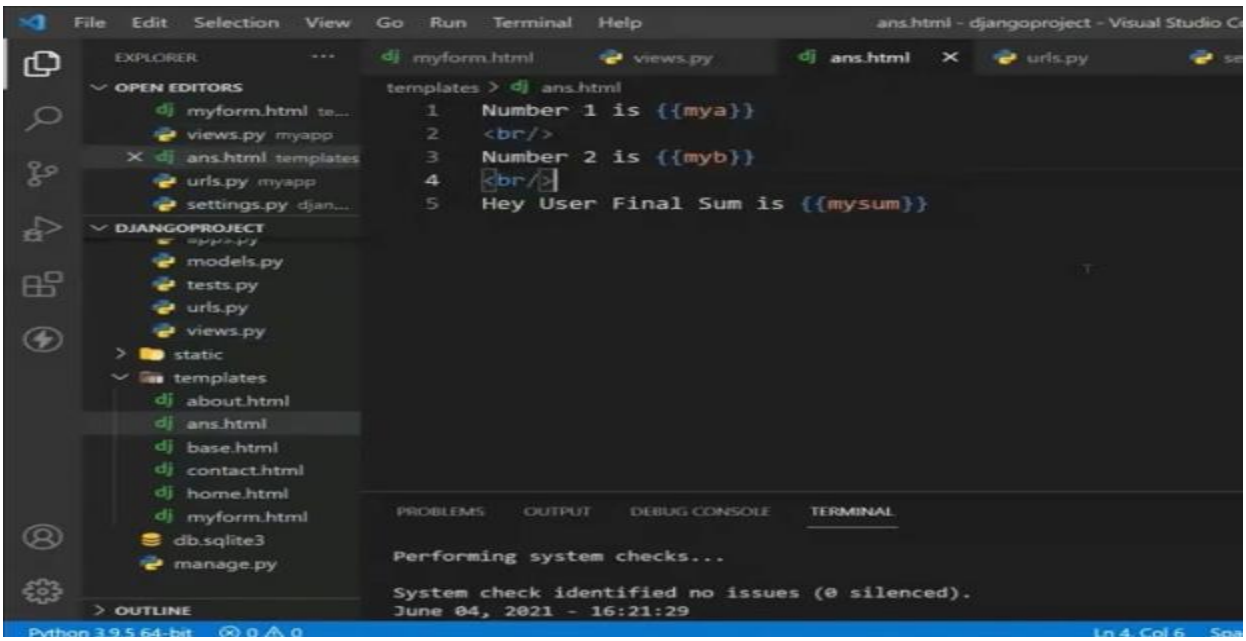
Urls.py



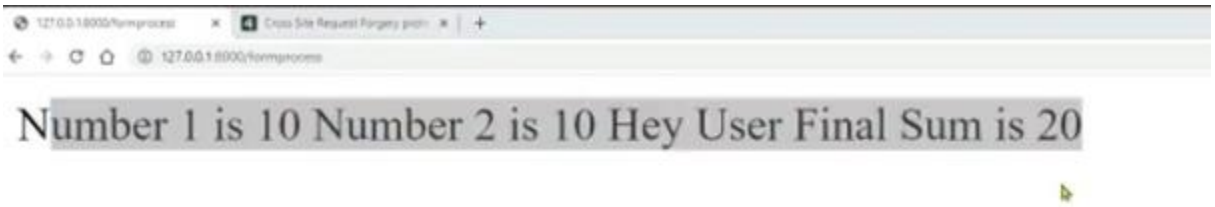
Views.py



Ans.html



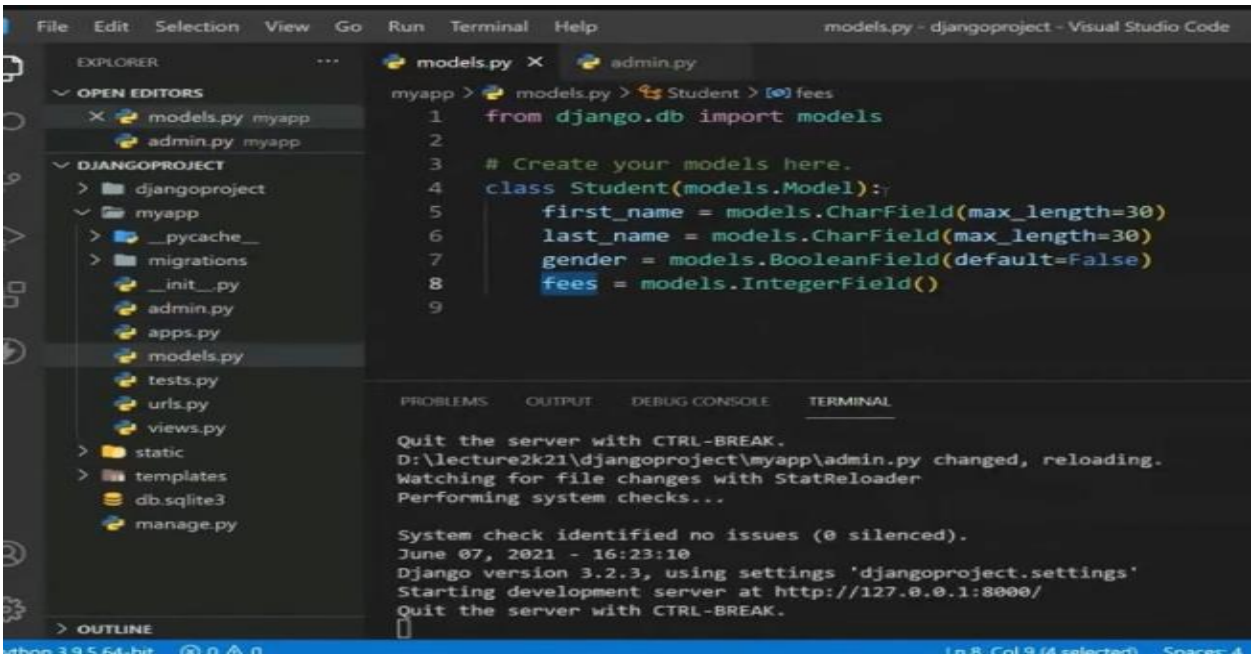
Output:-


Number 1 is 10 Number 2 is 10 Hey User Final Sum is 20

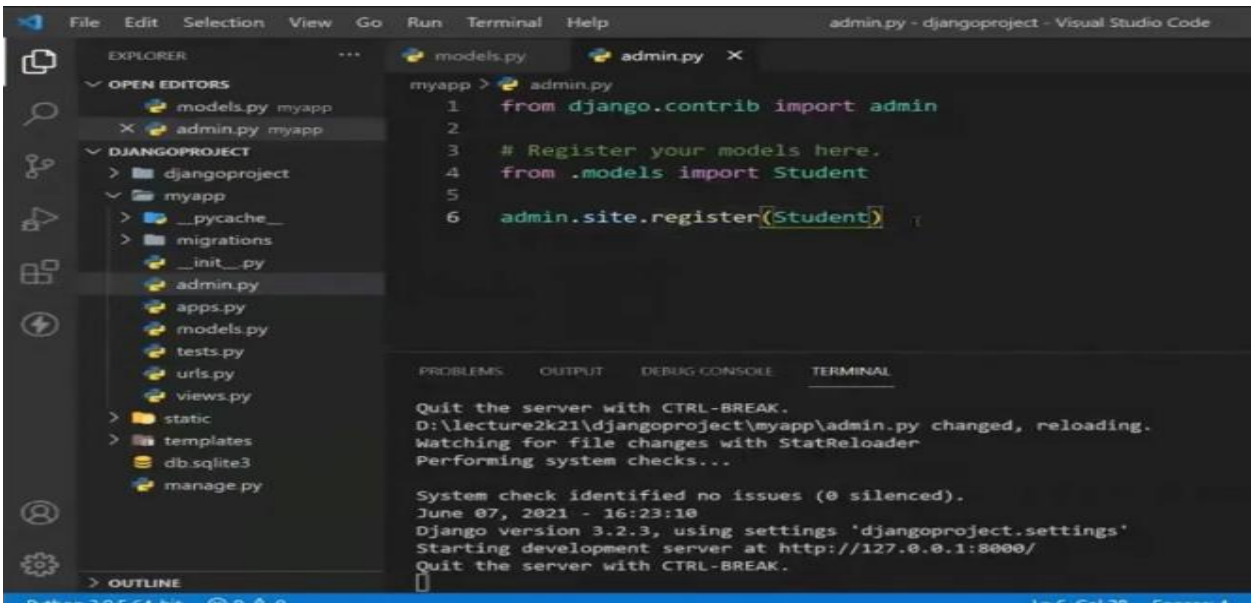
Task:10

How to Use Database in Django.

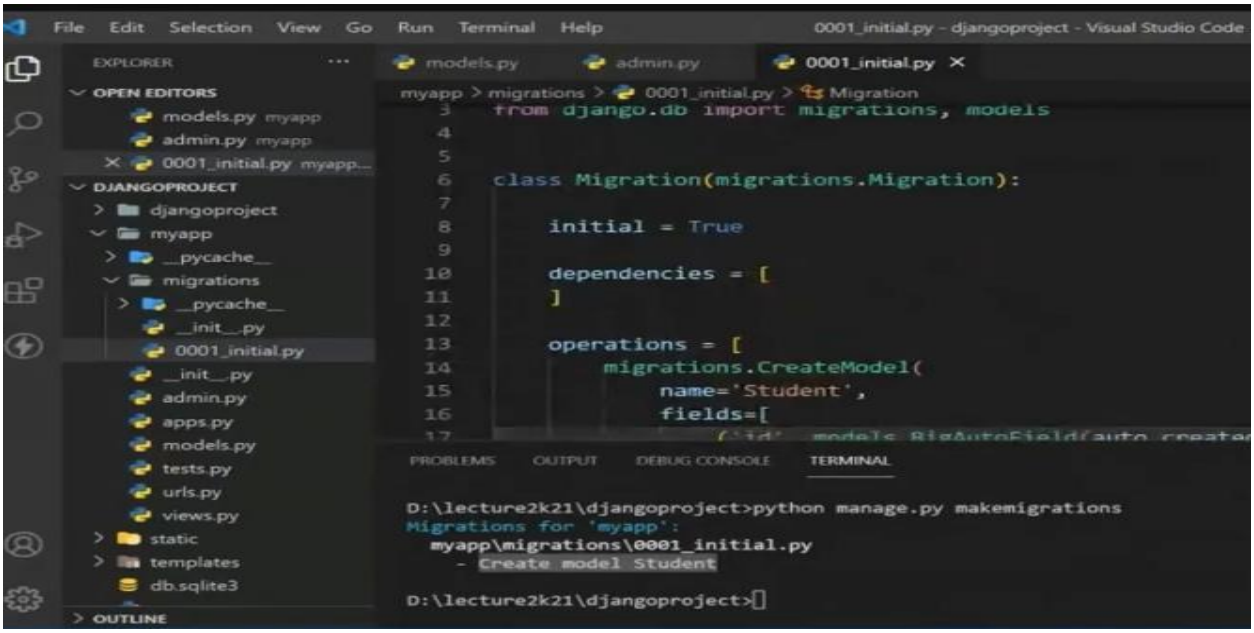
Model.py file

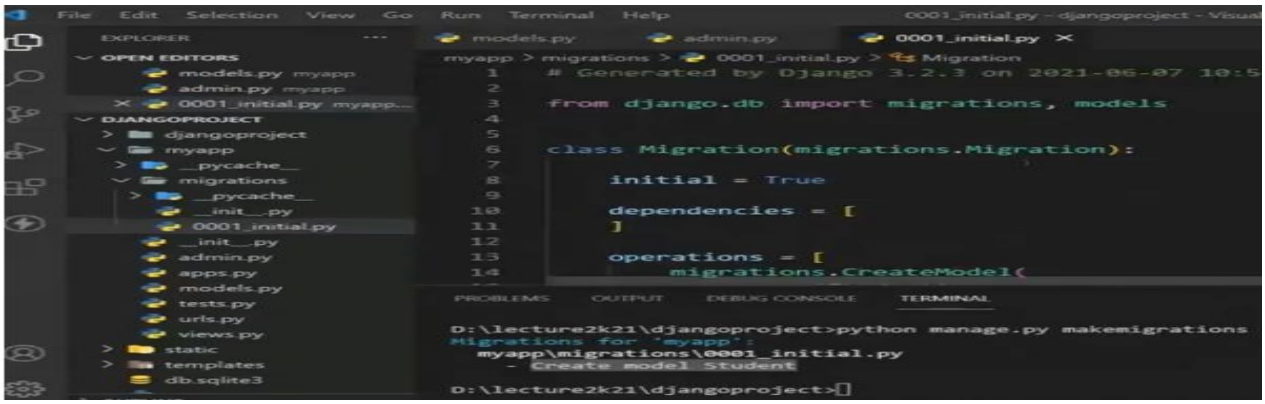


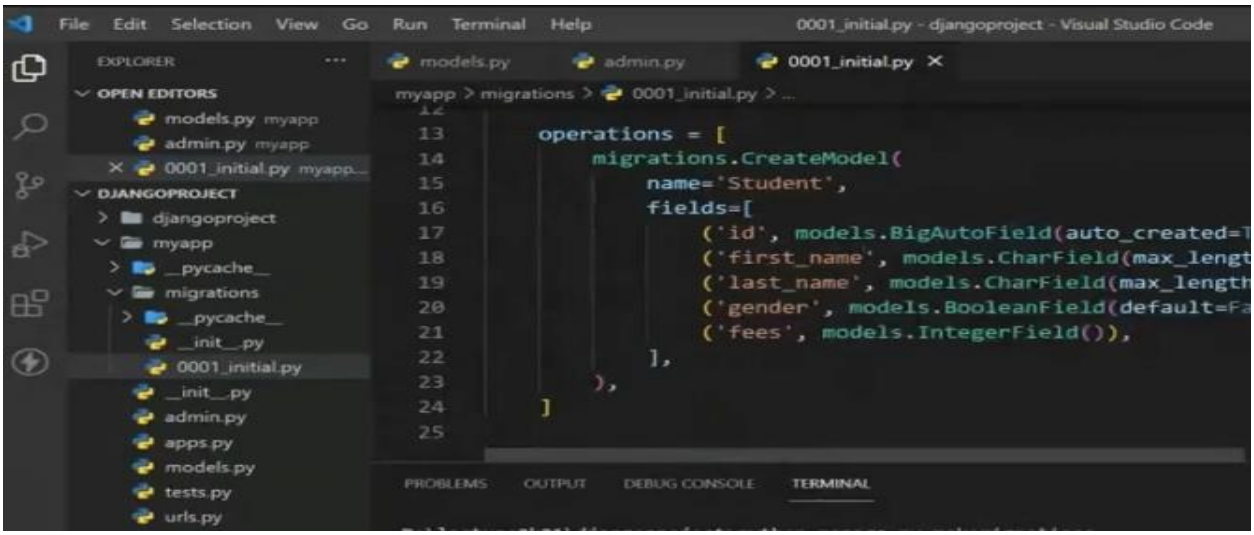
Admin file



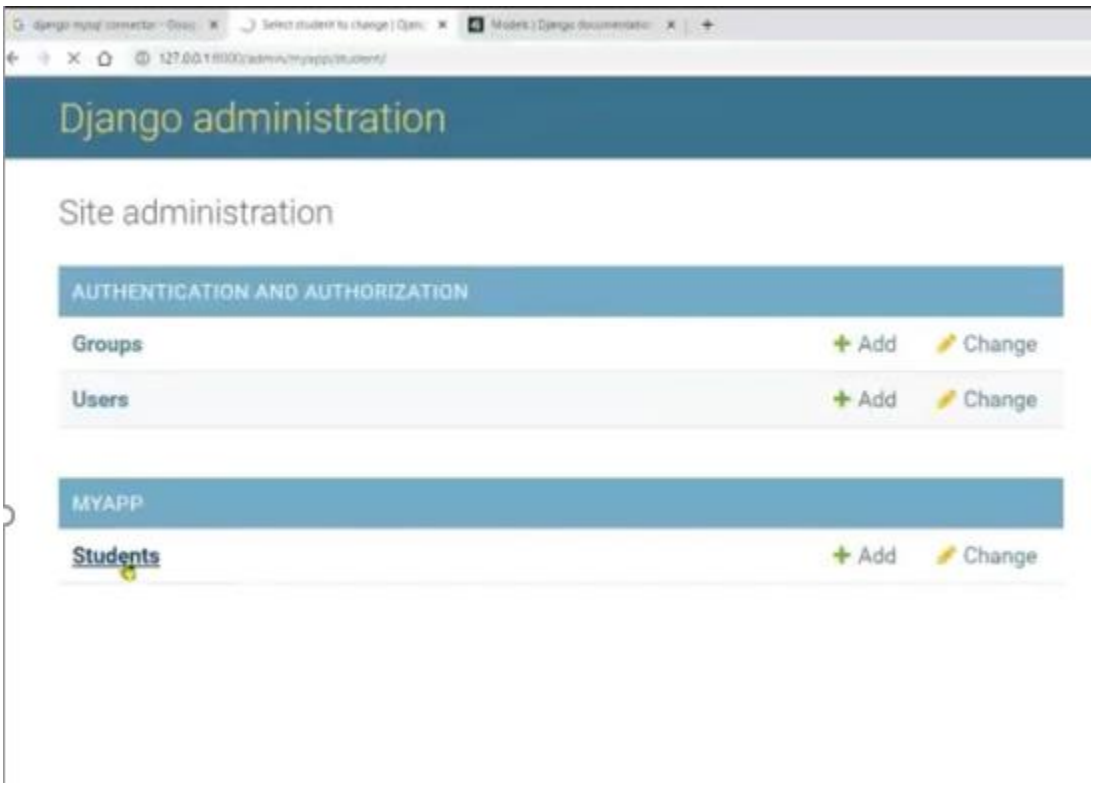
0001_initial

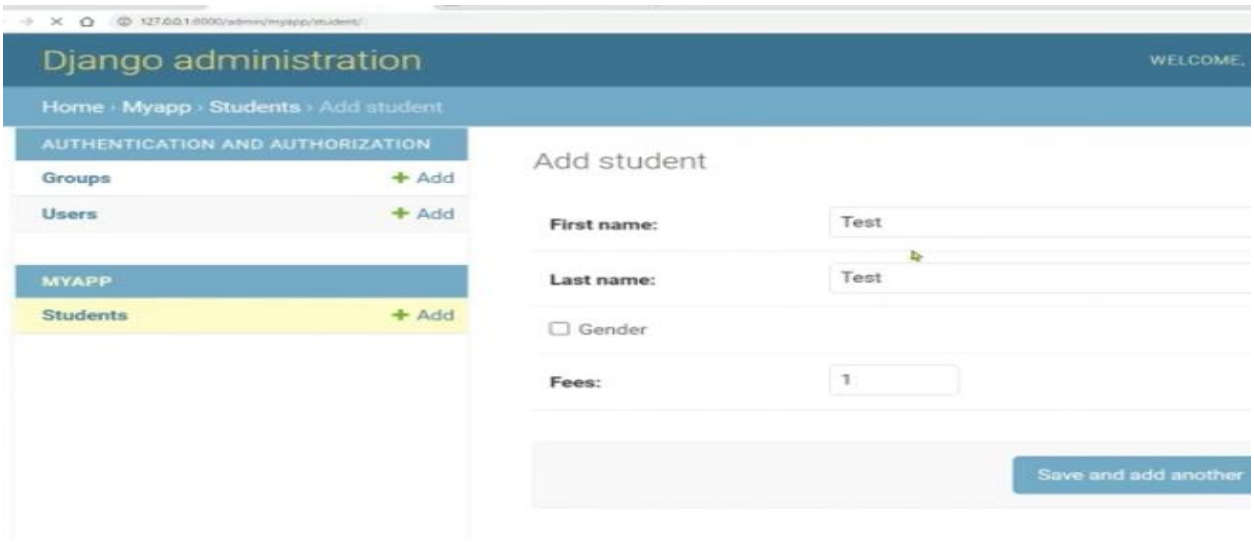





File Edit Selection View Go Run Terminal Help
0001_initial.py - djangoproject - Visual Studio Code
EXPLORER
OPEN EDITORS
models.py myapp
admin.py myapp
0001_initial.py myapp...
DJANGOPROJECT
djangoproject
myapp
__pycache__
migrations
__pycache__
__init__.py
0001_initial.py
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
models.py
admin.py
0001_initial.py
myapp > migrations > 0001_initial.py > ...
operations = [
migrations.CreateModel(
name='Student',
fields=[
('id', models.BigAutoField(auto_created=T
('first_name', models.CharField(max_lengt
('last_name', models.CharField(max_length
('gender', models.BooleanField(default=Fa
('fees', models.IntegerField()),
],
),
]
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

Output:-


Django administration
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups
+ Add
Change
Users
+ Add
Change
MYAPP
Students
+ Add
Change


Django administration
WELCOME,
Home › Myapp › Students › Add student
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
MYAPP
Students + Add
Add student
First name: Test
Last name: Test
Gender
Fees: 1
Save and add another