

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

<u>Index</u>	<u>Task Detail</u>	<u>Page Name</u>
1	Hello World	2
2	Variable and Data type	4
3	I/o function conditional statement	12
4	Function in Python	21
5	OOP's Concept	36
6	Install Django	50
7	Print Welcome to Django	55
8	In Website How to create pages in Django	58
9	How to Take value from user using Get and post method	60
10	How to Use Database in Django	64

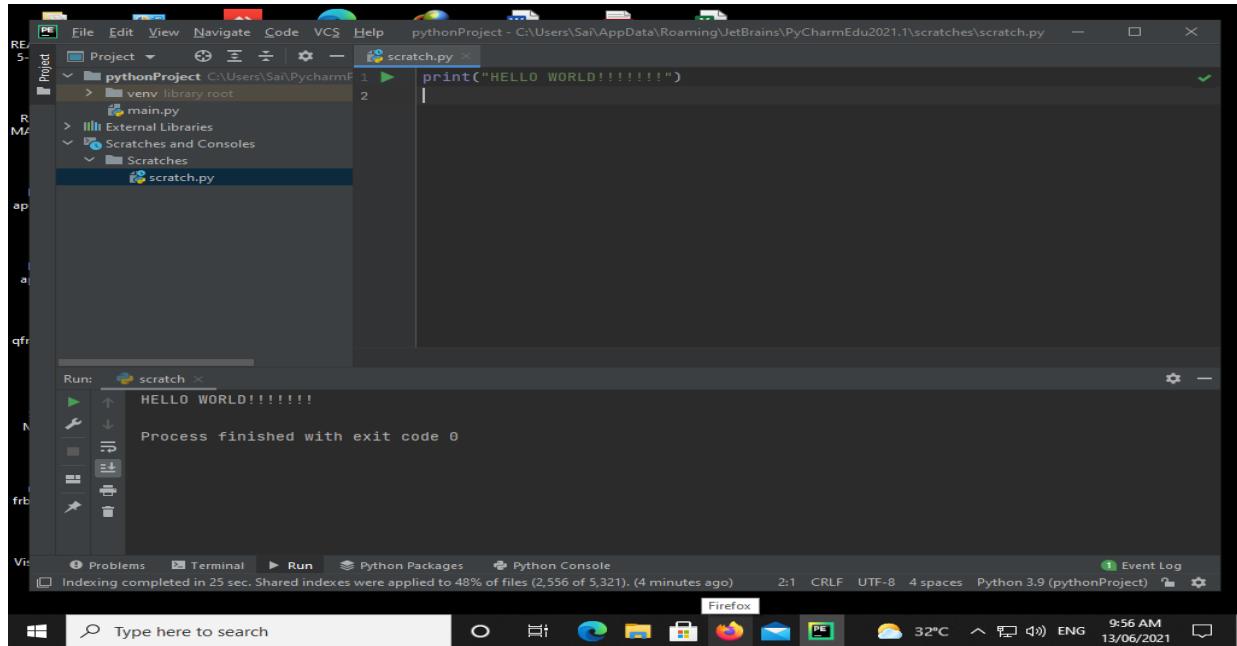
Task:1

Print hello world in python:-

Code:-

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

Output:-



Task:2

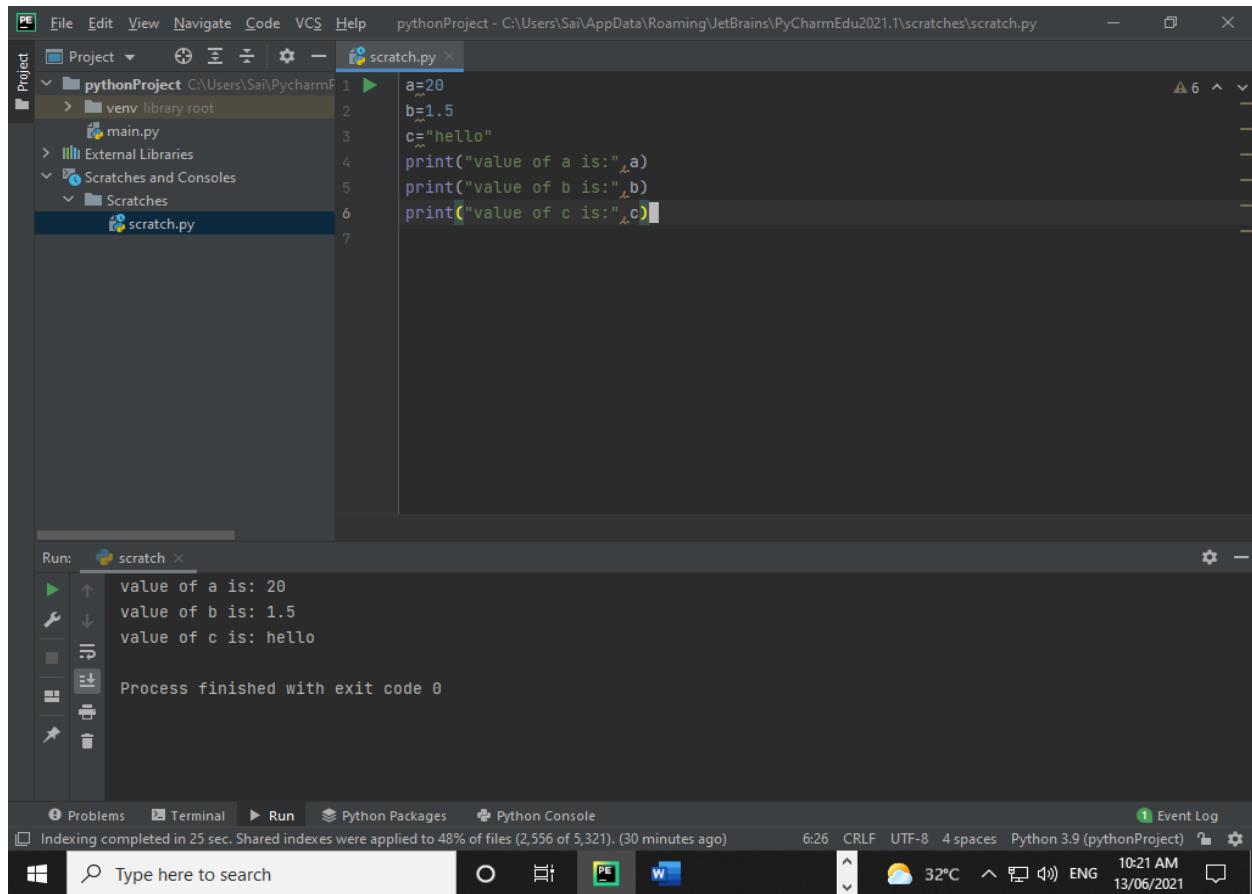
Variable and Datatype:-

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

Output:-



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, VCS, Help, and a title bar indicating 'pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py'. The left sidebar has a 'Project' view showing a 'pythonProject' folder containing 'venv', 'main.py', and 'Scratches' (which contains 'scratch.py'). The main code editor window displays the following Python code:

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

The 'Run' tool window at the bottom shows the output of running the script:

```
value of a is: 20
value of b is: 1.5
value of c is: hello

Process finished with exit code 0
```

The status bar at the bottom right shows indexing completion details, file encoding (CRLF), Python version (Python 3.9), and system information (32°C, ENG, 10:21 AM, 13/06/2021).

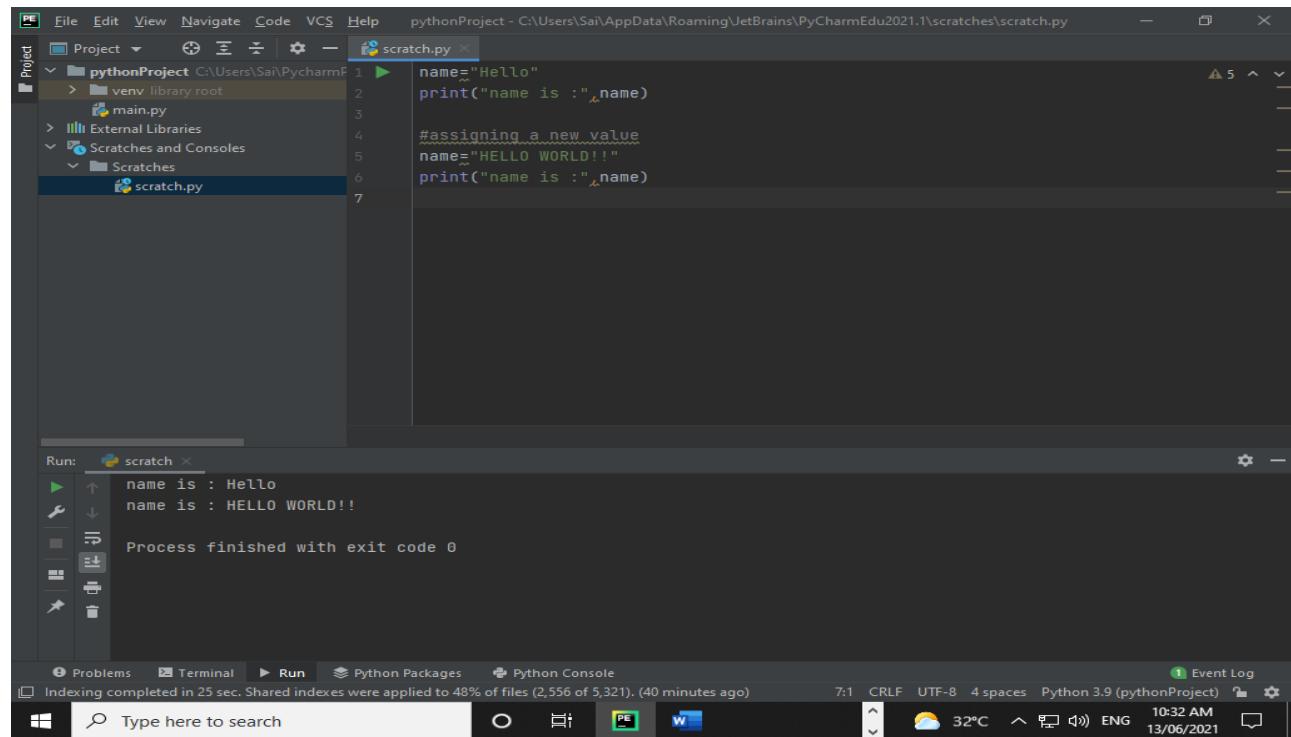
Changing the value of variable

Code:-

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

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

Output:-



The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, VCS, and Help. The title bar indicates the project is named 'pythonProject' and the file is 'scratch.py'. The left sidebar shows a project structure with a 'pythonProject' folder containing a 'venv' library root and a 'Scratches' folder with a 'Scratches' subfolder containing 'scratch.py'. The main code editor window displays the provided Python script. Below the editor is the 'Run' tool window, which shows the output of running the script. The output pane displays two lines of text: 'name is : Hello' and 'name is : HELLO WORLD!!', followed by 'Process finished with exit code 0'. At the bottom of the screen, the Windows taskbar is visible, showing the Start button, a search bar with the placeholder 'Type here to search', and several pinned application icons.

Assign multiple values:-

Code

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

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

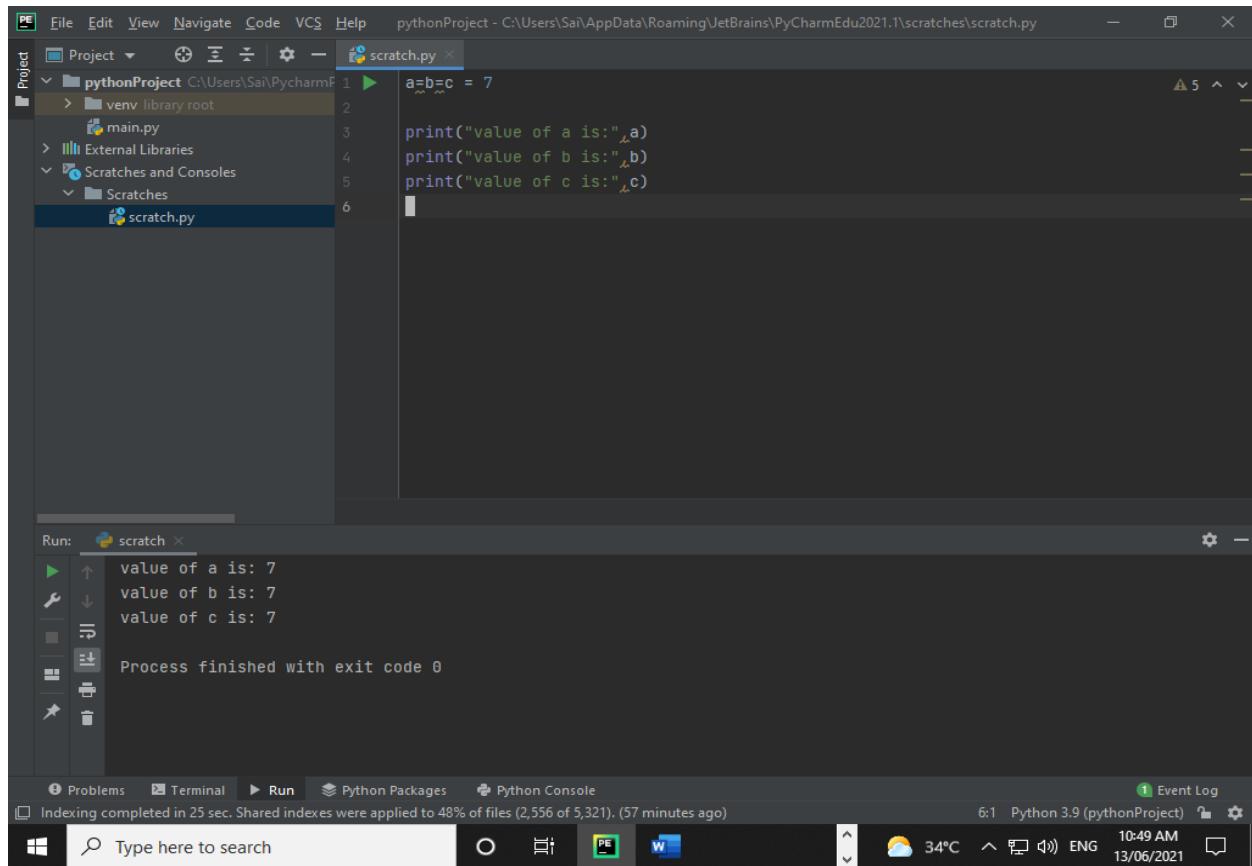
Output:-

The screenshot shows the PyCharm Edu interface with a dark theme. In the top navigation bar, the file 'scratch.py' is selected. The left sidebar displays a project structure with a 'pythonProject' folder containing 'venv', 'library root', 'main.py', and 'Scratches and Consoles' which contains 'Scratches' and 'scratch.py'. The main editor window shows the Python code: 'a, b, c = "orange", "banana", "cherry"\nprint(a)\nprint(b)\nprint(c)'. Below the editor is the 'Run' tool window, which has a dropdown set to 'scratch' and lists three runs: 'orange', 'banana', and 'cherry'. At the bottom of the tool window, it says 'Process finished with exit code 0'. The bottom status bar shows indexing completion details, file encoding (UTF-8), Python version (Python 3.9), and system information (32°C, ENG, 10:41 AM, 13/06/2021).

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:-



The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py
- Code Editor:** The file scratch.py contains the code:

```
a=b=c = 7  
  
print("value of a is:",a)  
print("value of b is:",b)  
print("value of c is:",c)
```
- Run Tab:** The "scratch" run configuration is selected, showing the output:

```
value of a is: 7  
value of b is: 7  
value of c is: 7  
  
Process finished with exit code 0
```
- Bottom Status Bar:** Shows indexing completion information: "Indexing completed in 25 sec. Shared indexes were applied to 48% of files (2,556 of 5,321). (57 minutes ago)". It also displays the Python version: "6:1 Python 3.9 (pythonProject)" and the current date and time: "10:49 AM 13/06/2021".

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:-

```
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))
```

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:-

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

print(name[2:5])

print(name[2:])

print(name + "Hello")
```

Run: scratch >
name is : HELLO WORLD
H
LLO
LLO WORLD
HELLO WORLDHello
Process finished with exit code 0

List:-

Code:

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

print(list1)
```

Output:-

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

print(list1)
```

Run: scratch >
[1, 10.2, 'Hello World']
Process finished with exit code 0

Tuple:-

Code:

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

print (tuple1)
```

Output:-

A screenshot of the PyCharm IDE interface. The top window shows a code editor with a file named 'scratch.py' containing the following Python code:

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

The bottom window shows the 'Run' tab with the output of the code execution:

```
(10, 20, 30, 'Hello', 50, 60, 'world', 70)
Process finished with exit code 0
```

The status bar at the bottom right indicates the environment is Python 3.9 (pythonProject), the date is 13/06/2021, and the time is 11:35 AM.

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:-

A screenshot of the PyCharm IDE interface. The top window shows a code editor with a file named 'scratch.py' containing the following Python 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'])
```

The bottom window shows the 'Run' tab with the output of the code execution:

```
<class 'dict'>
d[1] = Hello
d[2] = Wolrd
d['key']= 20
Process finished with exit code 0
```

The status bar at the bottom right indicates the environment is Python 3.9 (pythonProject), the date is 13/06/2021, and the time is 11:35 AM.

Task:-3

I/o function conditional statement

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

Input([prompt])

Input:-

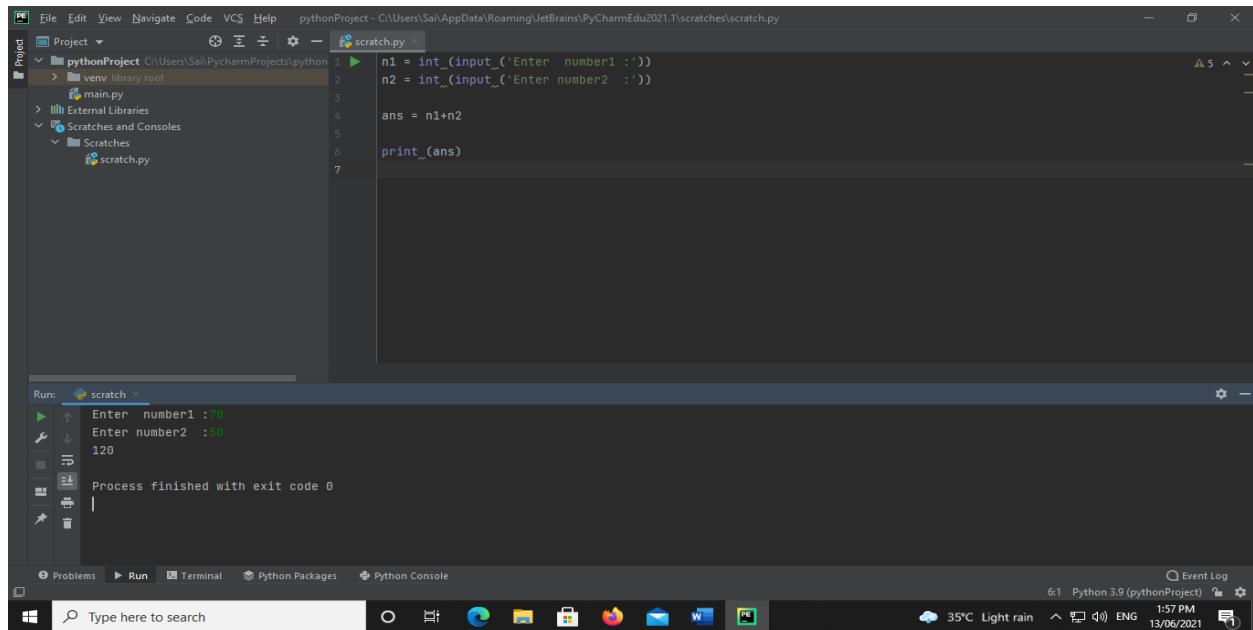
Code:

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

ans = n1+n2

print (ans)
```

Output:-



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, VCS, Help, and pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py. The Project tool window on the left shows a file tree with a project named 'pythonProject' containing a 'venv library root' and a 'Scratches and Consoles' section with a 'Scratches' folder containing 'scratch.py'. The main code editor window displays the following Python code:

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

ans = n1+n2

print(ans)
```

The 'Run' tab at the bottom shows the execution results. It prompts for 'Enter number1 : 20' and 'Enter number2 : 50', then displays the output '120'. Below the output, it says 'Process finished with exit code 0'. The status bar at the bottom right shows the date and time as 13/06/2021 1:57 PM.

Conditional Statement:

If Statement

Syntax:-

If condition:

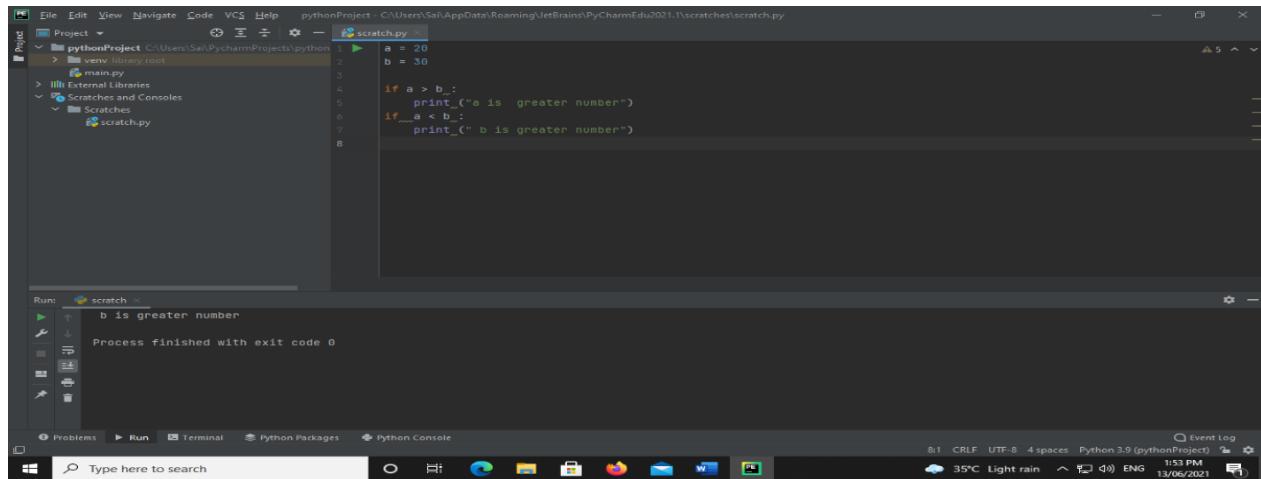
Statement(s)

Code:

```
a = 20
b = 30

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

Output:-



```
a = 20
b = 30

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

Run: scratch >
b is greater number
Process finished with exit code 0

Ifelse Statement

Syntax:-

If condition:

Statements to be executed

Else:

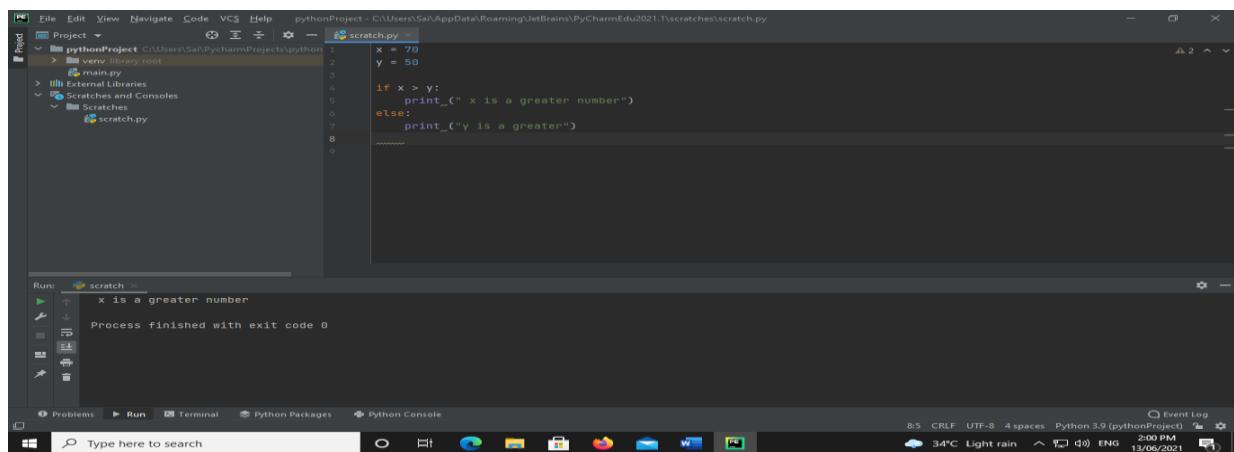
Statements to be executed

Code:-

```
x = 70
y = 50

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

Output:-



The screenshot shows the PyCharm IDE interface. The project structure on the left includes a 'pythonProject' folder with 'main.py' and 'Scratches' containing 'scratch.py'. The code in 'scratch.py' is as follows:

```
x = 70
y = 50

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

In the 'Run' tool window at the bottom, the output shows:

```
x is a greater number
Process finished with exit code 0
```

The status bar at the bottom right indicates the time as 8:55, weather as 34°C Light rain, and system information.

If.... elif ...else Statement

Syntax:-

If condition:

 Statements to be executed

elif:

 Statements to be executed

else:

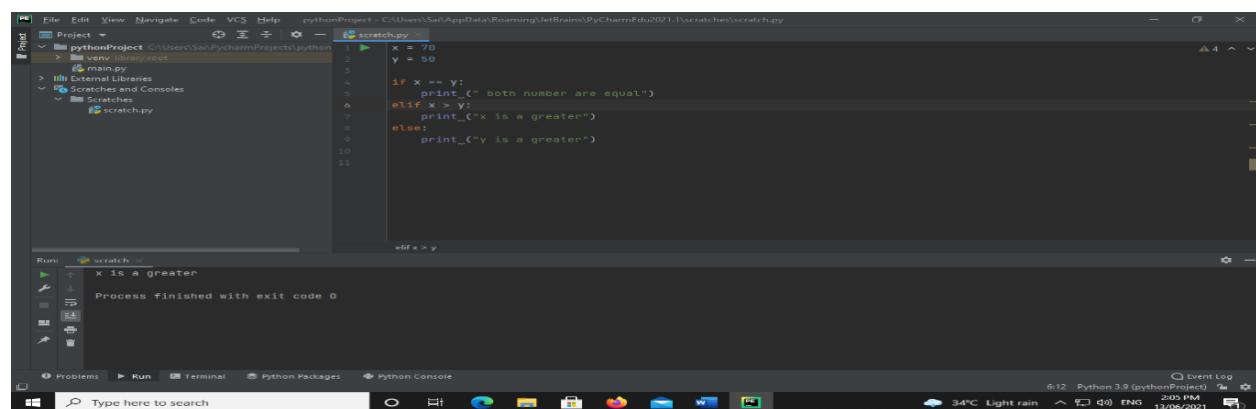
 Statements to be executed

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:-



The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, VCS, Help, and a project dropdown for 'pythonProject'. Below the navigation bar is the Project tool window showing a file named 'scratch.py'. The code editor contains the following Python script:

```
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")
```

The Run tool window at the bottom shows the output of the script: "x is a greater". The status bar at the bottom right indicates the time as 6:12 PM, the date as 13/06/2021, and the system temperature as 34°C.

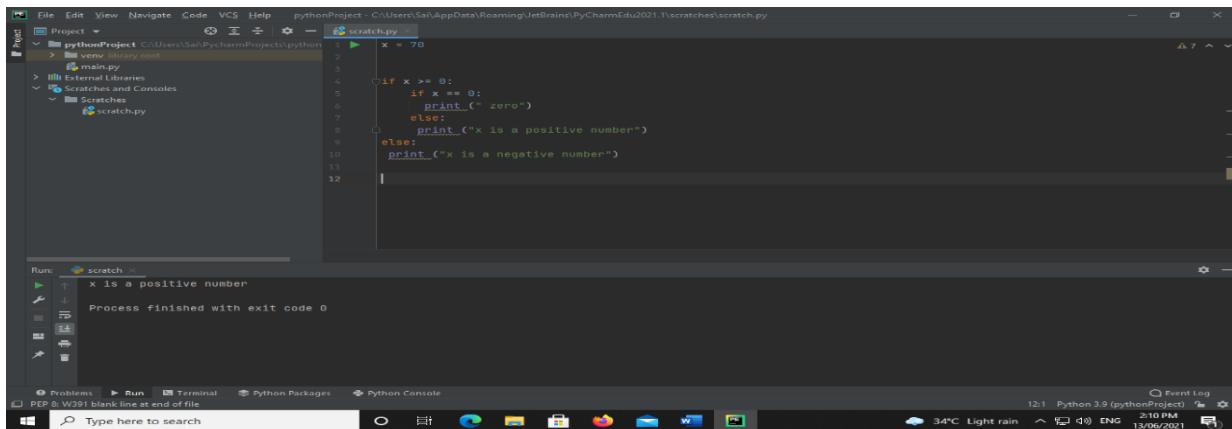
Nested if statement

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:-



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, VCS, Help, and pythonProject - C:\Users\Sai\AppData\Roaming JetBrains\PyCharm Edu 2021.1\scratches\scratch.py. The Project tool window on the left shows a file named scratch.py under a folder named pythonProject. The code editor window displays the provided Python script. The bottom Run tab shows the terminal output: "x is a positive number". The status bar at the bottom right indicates the time as 12:11, Python 3.9 (pythontests), and the date as 13/06/2021.

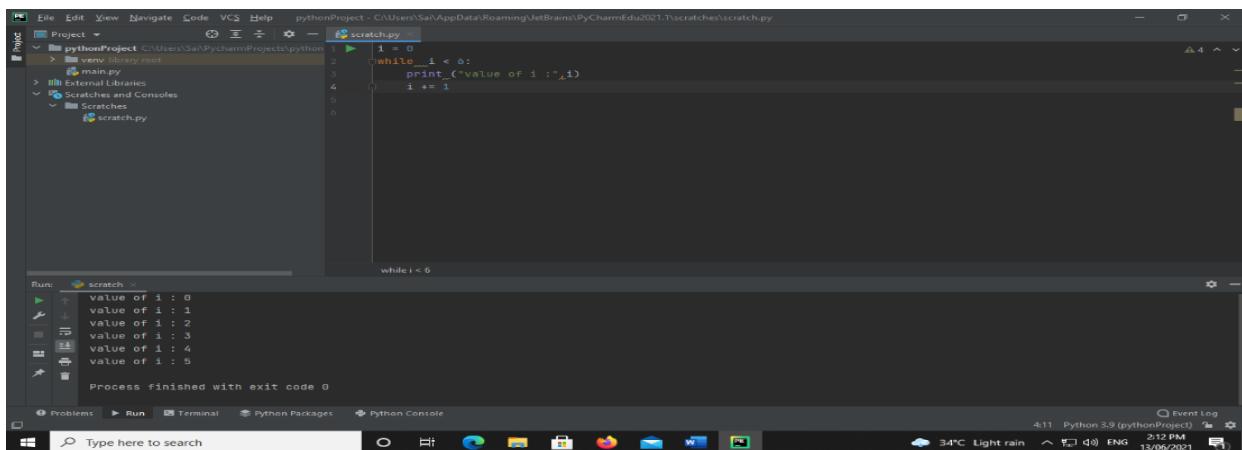
Loop

while loop

code:

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

Output:-



The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, VCS, Help, and a path: pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py. The left sidebar has a Project tree with a single file 'scratch.py' under 'pythonProject'. The main code editor window contains the following Python code:

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

Below the code editor is a 'Run' tool window titled 'scratch'. It shows the output of the run: 'value of i : 0', 'value of i : 1', 'value of i : 2', 'value of i : 3', 'value of i : 4', and 'value of i : 5'. At the bottom of the Run window, it says 'Process finished with exit code 0'. The bottom status bar shows the system tray with icons for battery, signal, and time (13/06/2021 2:12 PM). The taskbar at the bottom of the screen also displays various application icons.

for loop

Code:

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

Output:-

```
for i in 'ARMY':
    print_(value := i)
```

```
for i in 'ARMY'
  value : A
  value : R
  value : M
  value : Y
Process finished with exit code 0
```

the range () function

Code:

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

Output:-

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

```
for y in range (3 , 5)
  first range : 0
  first range : 1
  first range : 2
  first range : 3
  second range : 3
  second range : 4
Process finished with exit code 0
```

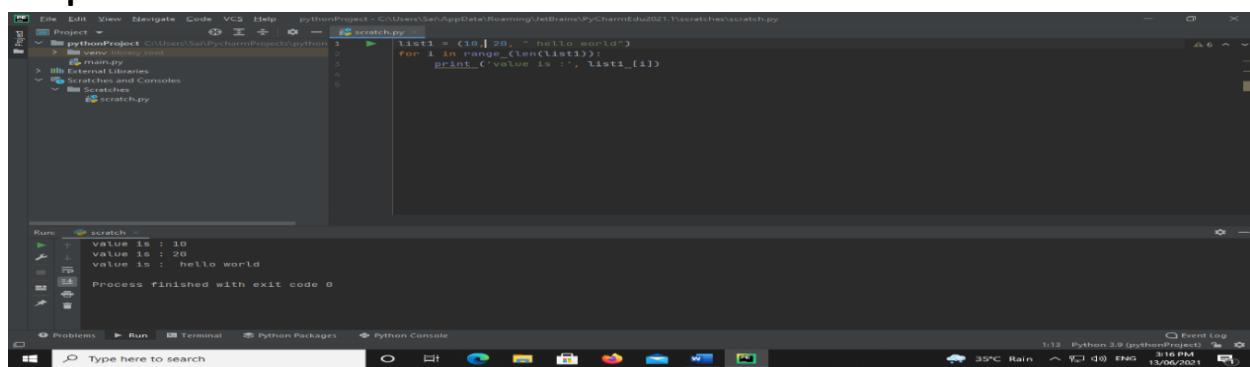
loop with else

Code:

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

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

Output:-



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

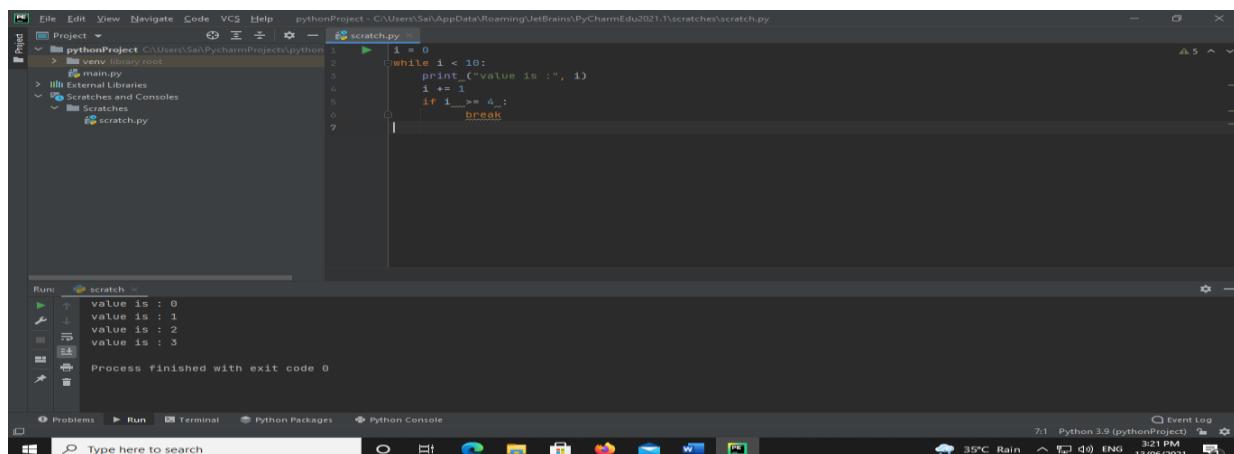
```
Value is : 10  
Value is : 20  
Value is : hello world  
Process finished with exit code 0
```

break statement

Code:

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

Output:-



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

```
value is : 0  
value is : 1  
value is : 2  
value is : 3  
Process finished with exit code 0
```

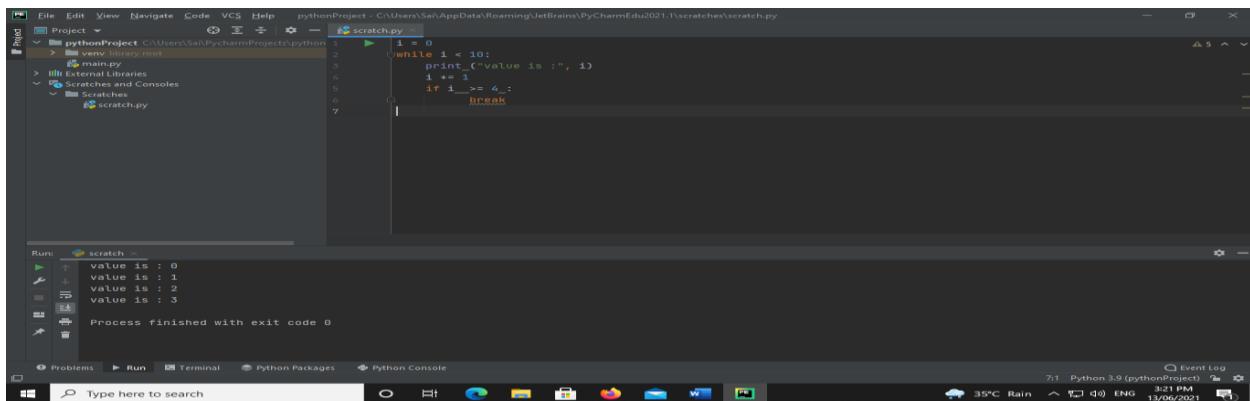
Continue

Code:

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

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

Output:-



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

Run → scratch

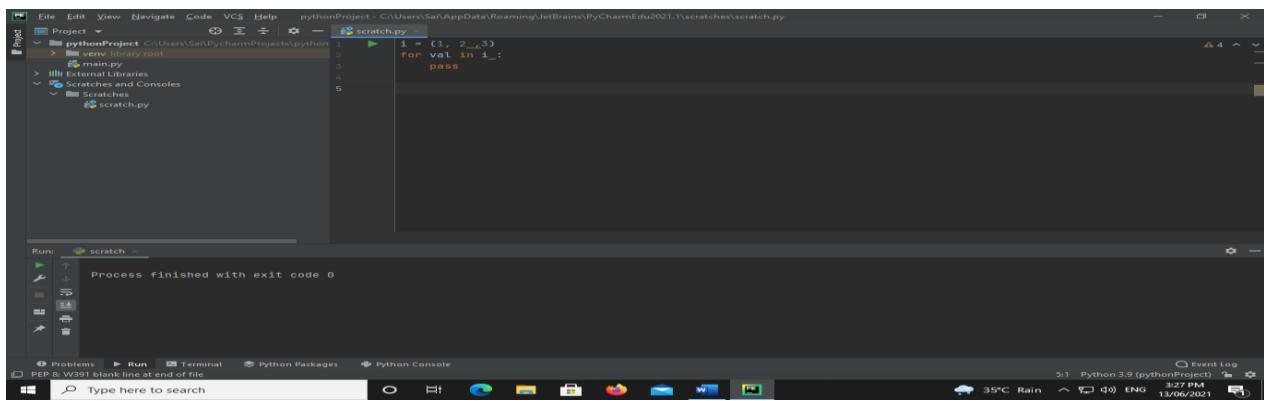
```
Value is : 0
Value is : 1
Value is : 2
Value is : 3
Process finished with exit code 0
```

Pass

Code:

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

Output:-



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

Run → scratch

```
Process finished with exit code 0
```

Task:4

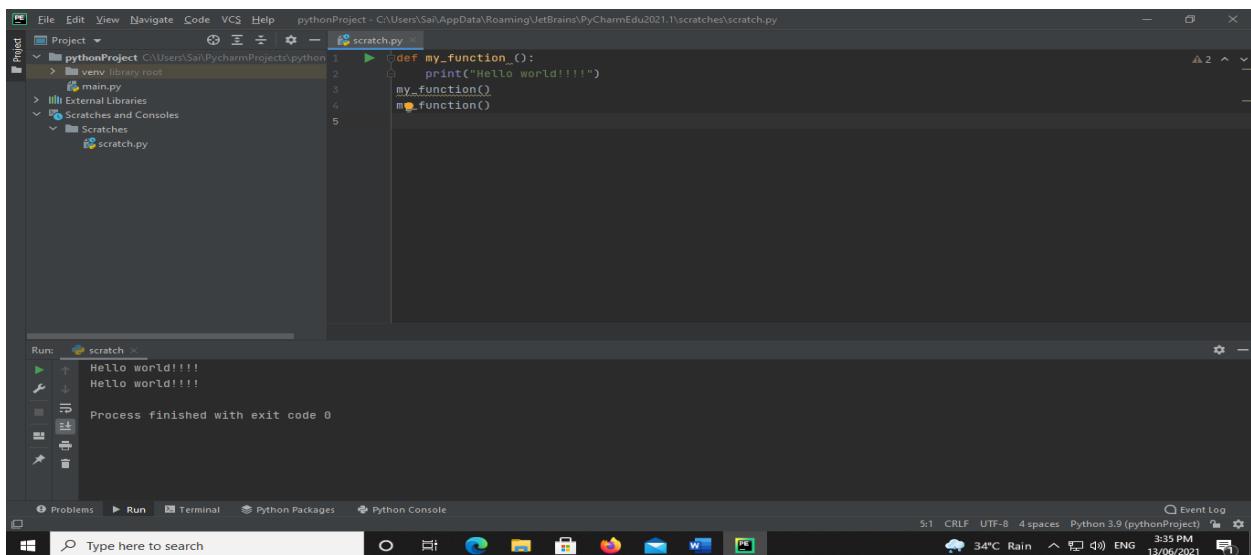
Function in python

- 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:-

The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, VCS, Help, and pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py. The left sidebar shows a Project tree with a single file named scratch.py. The code editor contains the following Python code:

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

The Run tab at the bottom has the command "scratch" and the output shows the result of the function call:

```
name is : nature
Process finished with exit code 0
```

The status bar at the bottom right indicates the system is 34°C Rain, the time is 3:37 PM, and the date is 13/06/2021.

Function with return Statement

Code:

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

Output:-

The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, VCS, Help, and pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py. The left sidebar shows a Project tree with a single file named scratch.py. The code editor contains the following Python code:

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

The Run tab at the bottom has the command "scratch" and the output shows the result of the function call:

```
Hello world
Process finished with exit code 0
```

The status bar at the bottom right indicates the system is 33°C Rain, the time is 6:11 PM, and the date is 13/06/2021.

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:-

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

Run: scratch ×
Process finished with exit code 0

Python function argument:-

Code:

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

Output:-

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

Run: scratch ×
30
12
Process finished with exit code 0

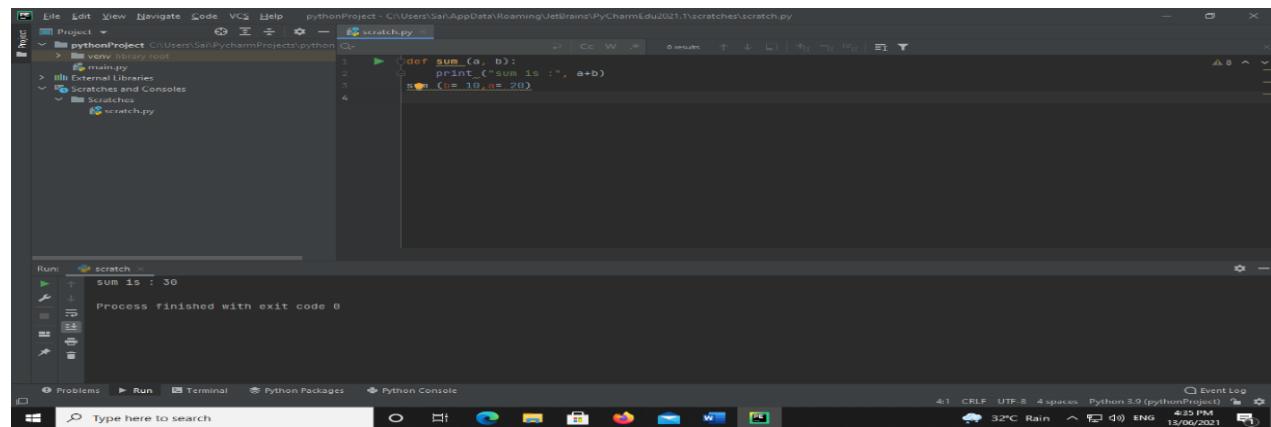
Python keyword arguments:

- 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:-



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

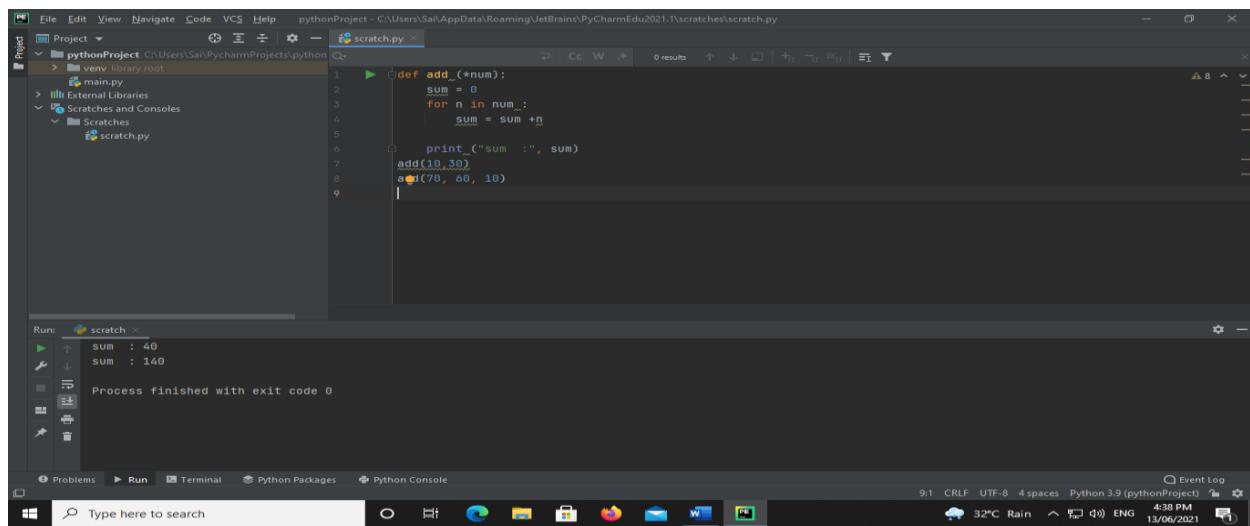
Run → scratch → sum is : 30
Process finished with exit code 0

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:-



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

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

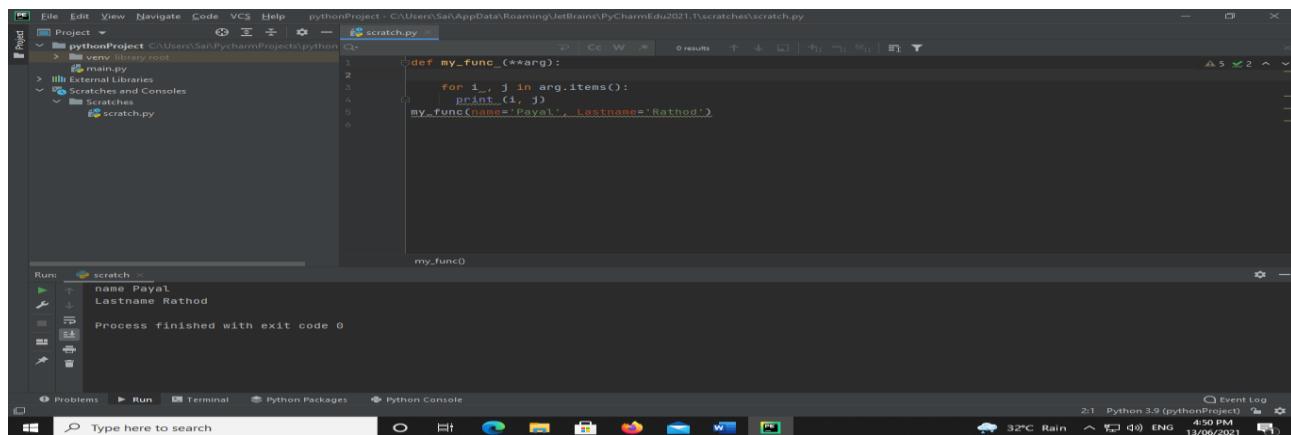
Run: scratch
 sum : 40
sum : 130
Process finished with exit code 0

Keyword arguments:-

Code:

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

Output:-



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

Run: scratch
 name Payal
Lastname Rathod
Process finished with exit code 0

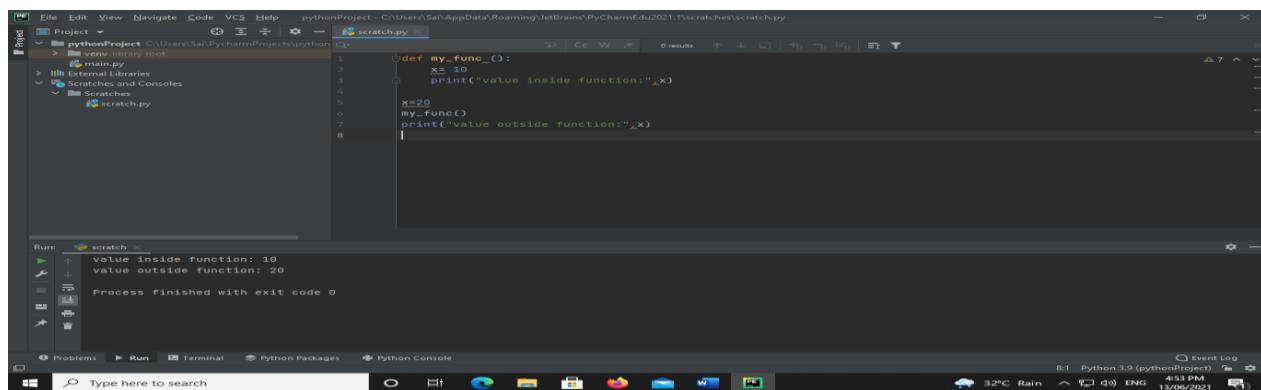
Scope of variable:

- 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 screenshot shows the PyCharm IDE interface. The code editor window displays the script 'scratch.py' with the following content:

```
def my_func ():  
    x= 10  
    print("value inside function:",x)  
  
x=20  
my_func()  
print("value outside function:",x)
```

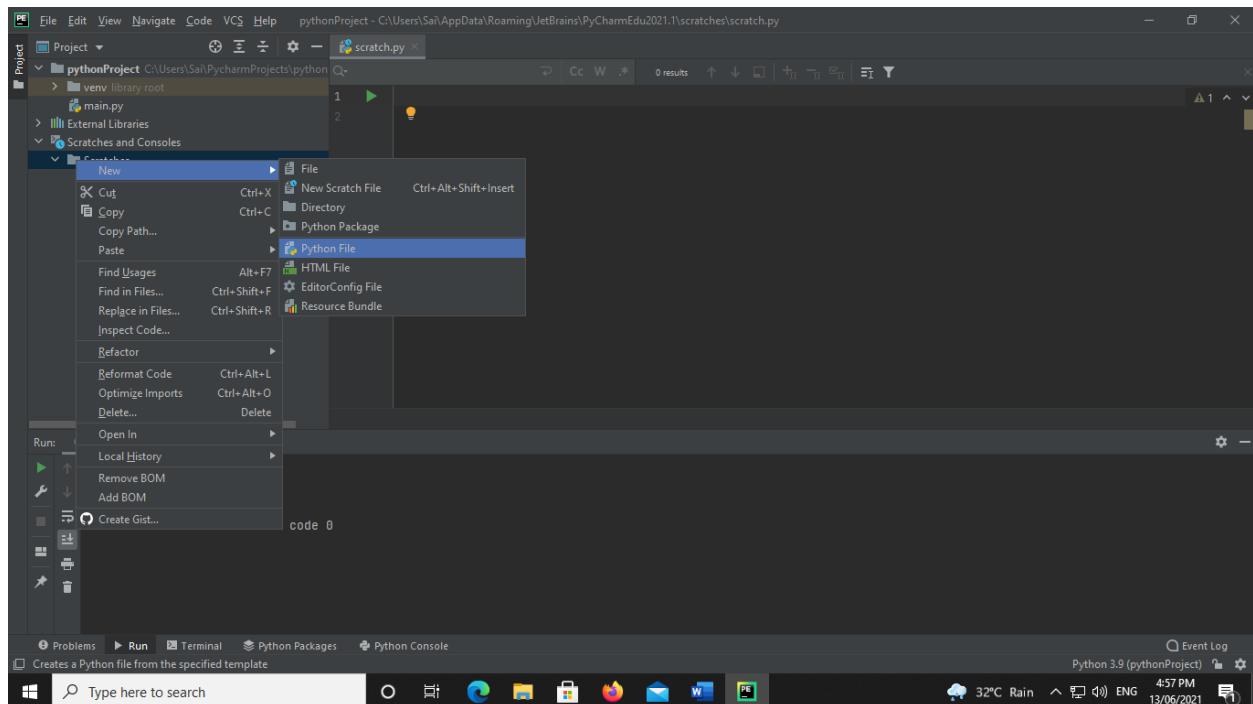
The 'Run' tool window at the bottom shows the execution results:

```
value inside function: 10  
value outside function: 20  
Process finished with exit code 0
```

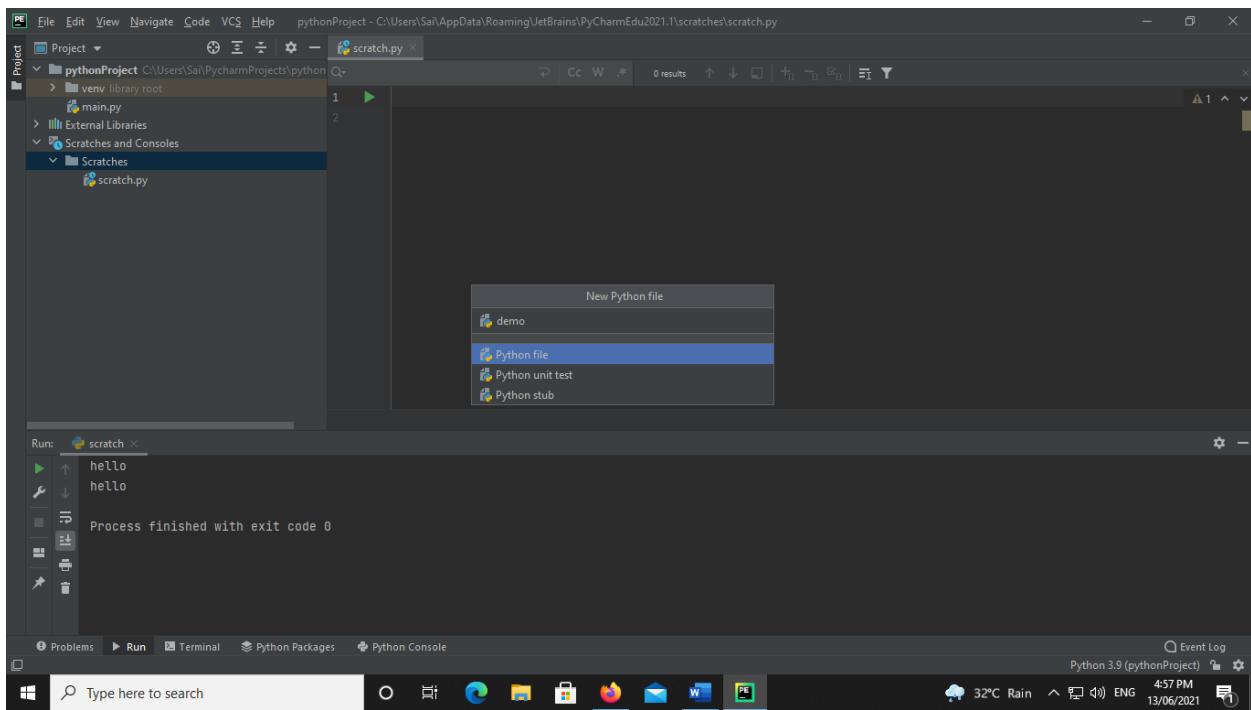
The status bar at the bottom right indicates the environment: Python 3.9 (pythonProject), 8:1, 32°C Rain, ENG, 15:51 PM.

The Import Statement:-

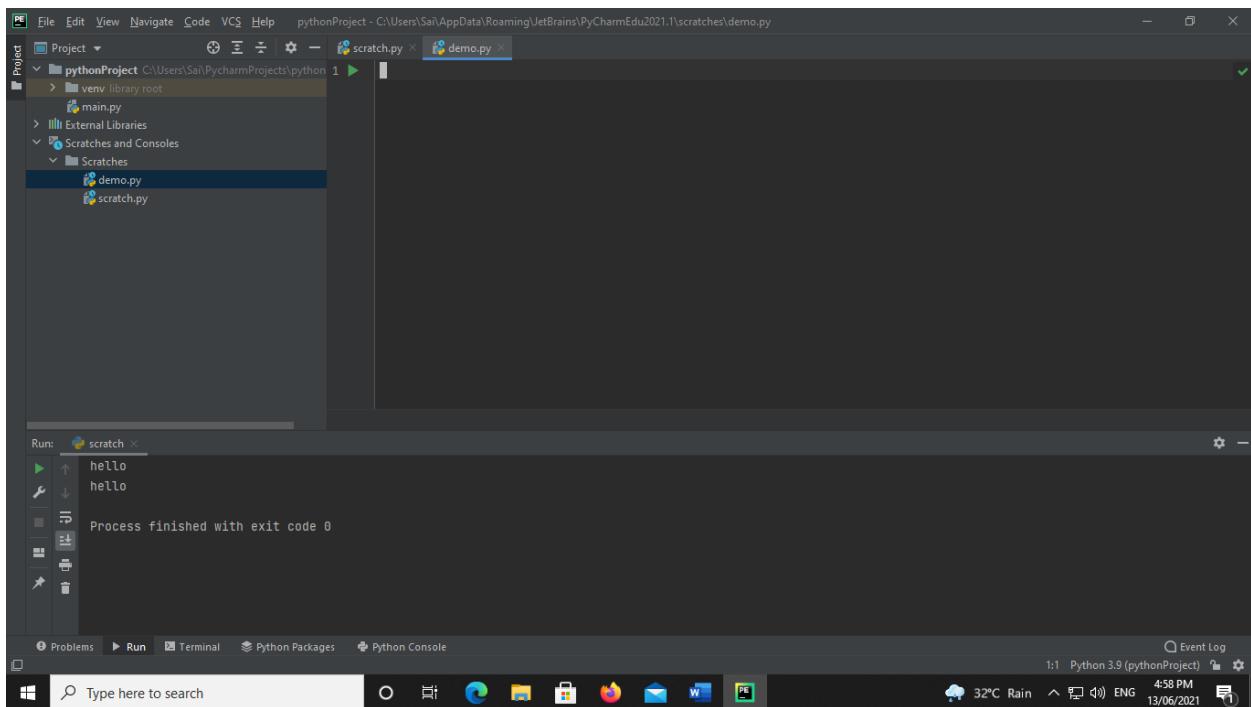
Create a new file by right clicking on Scratches folder:—



Give file name



File is created



Scratch.py

The screenshot shows the PyCharm IDE interface. In the top navigation bar, the project is named 'pythonProject' and the file is 'demo.py'. The code editor contains the following Python code:

```
def my_function(name):
    return name
```

In the bottom run panel, the output of the script is shown:

```
hello
hello
Process finished with exit code 0
```

The status bar at the bottom right indicates the time is 4:59 PM, the weather is 32°C Rain, and the system language is ENG.

Demo.py

The screenshot shows the PyCharm IDE interface. In the top navigation bar, the project is named 'pythonProject' and the file is 'demo.py'. The code editor contains the following Python code:

```
import scratch
name = scratch.my_function("hello world")
print(name)
```

In the bottom run panel, the output of the script is shown:

```
hello world
Process finished with exit code 0
```

The status bar at the bottom right indicates the time is 4:12 PM, the weather is 32°C Rain, and the system language is ENG.

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:-

```
pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py
Project: pythonProject C:\Users\Sai\PycharmProjects\python
File Edit View Navigate Code VCS Help pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py
scratch.py
1 x= 7
2 y= 5
3 print('x + y =', x+y)
4 print('x - y =', x-y)
5 print('x * y =', x*y)
6 print('x / y =', x/y)
7 print('x // y =', x//y)
8 print('x ** y =', x**y)

Run: scratch
x + y = 12
x - y = 2
x * y = 35
x / y = 1.4
x // y = 1
x ** y = 16807
Process finished with exit code 0

8:22 Python 3.9 (pythonProject) Event Log
32°C Rain 5:06 PM 13/06/2021
```

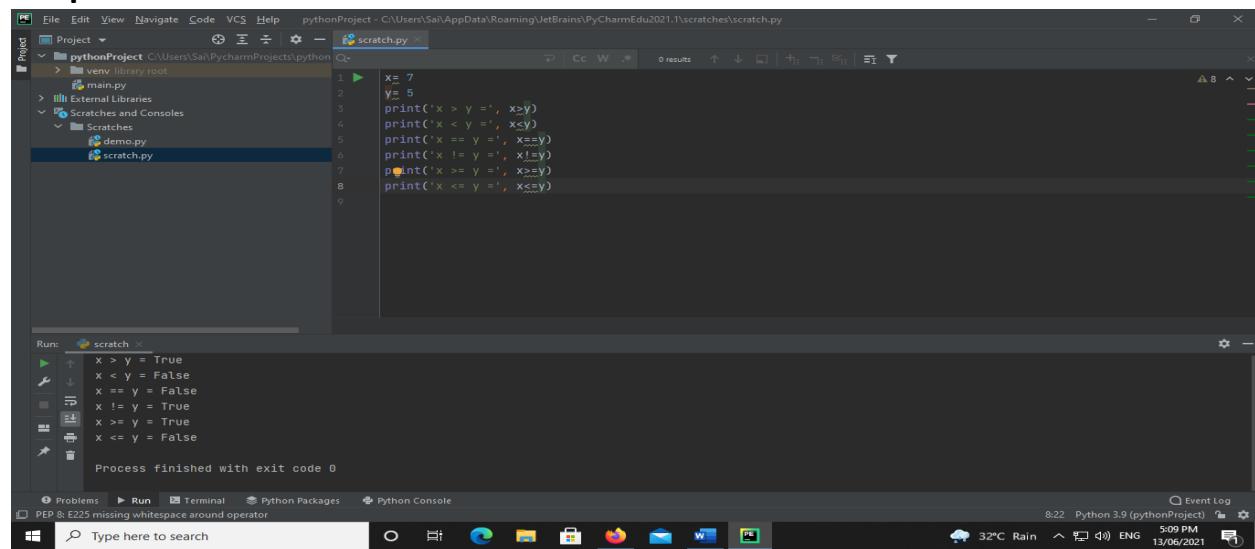
- Comparison Operators

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:-



The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py
- Code Editor:** The file scratch.py contains the provided Python code for testing comparison operators.
- Run Tab:** The "Run" tab is selected, showing the output of the code execution.
- Output:**

```
x > y = True
x < y = False
x == y = False
x != y = True
x >= y = True
x <= y = False
```
- Status Bar:** Shows the time (8:22), weather (32°C Rain), and date (13/06/2021).

- **Logical Operators**

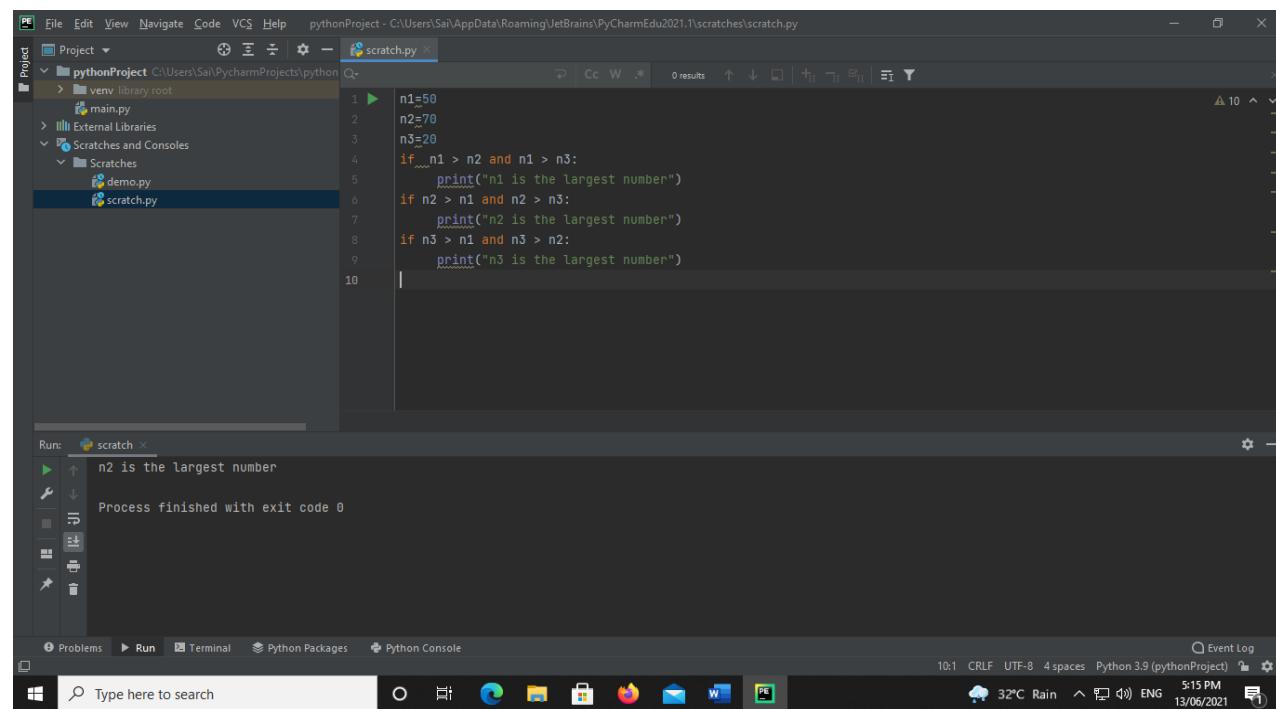
Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

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:-



The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject
- File:** scratch.py
- Code Content:**

```
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")
```
- Run Output:**

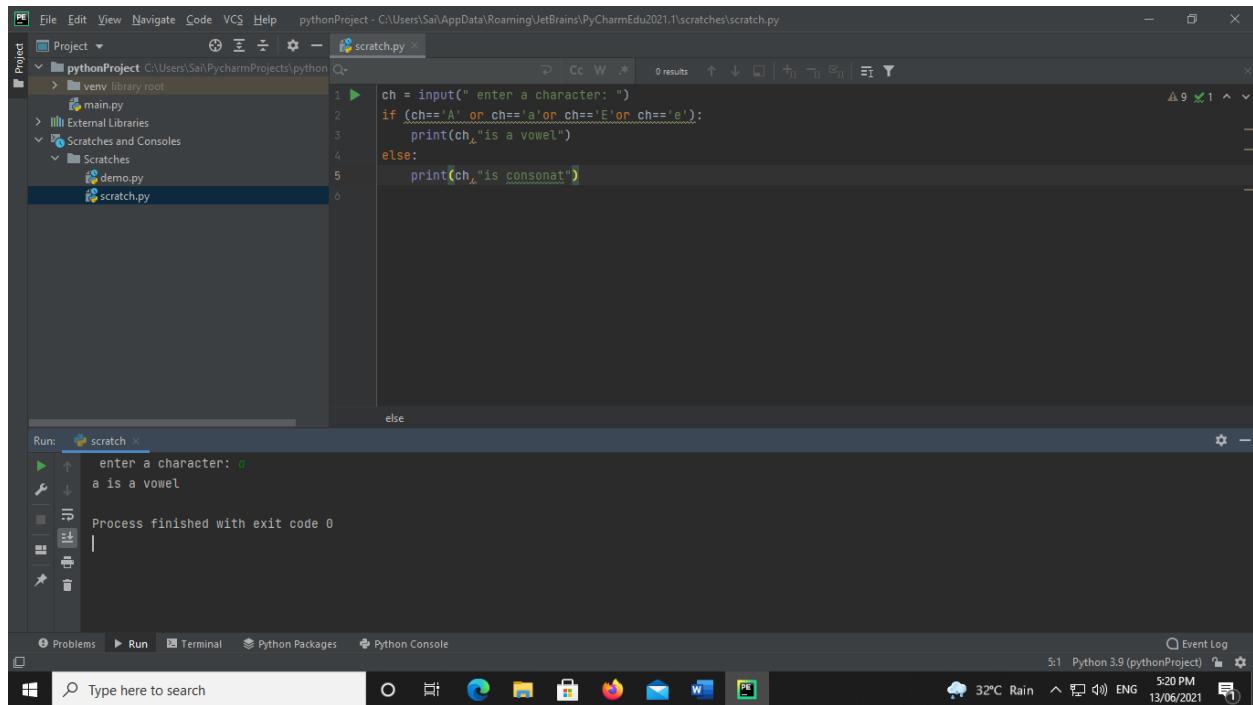
```
n2 is the largest number
Process finished with exit code 0
```
- Bottom Status Bar:**
 - 10:1 CRLF UTF-8 4 spaces Python 3.9 (pythonProject)
 - 32°C Rain 5:15 PM 13/06/2021

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 consonant")
```

Output:-



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, VCS, Help, and a path indicating the project is 'pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py'. The main window has a 'Project' tool window on the left showing files like 'main.py', 'Scratches', and 'scratch.py'. The central code editor window displays the provided Python script. Below the editor is the 'Run' tool window, which shows the execution results: 'enter a character: a', followed by 'a is a vowel', and 'Process finished with exit code 0'. At the bottom, there's a search bar, a toolbar with various icons, and a status bar showing weather (32°C Rain), time (5:20 PM), and date (13/06/2021).

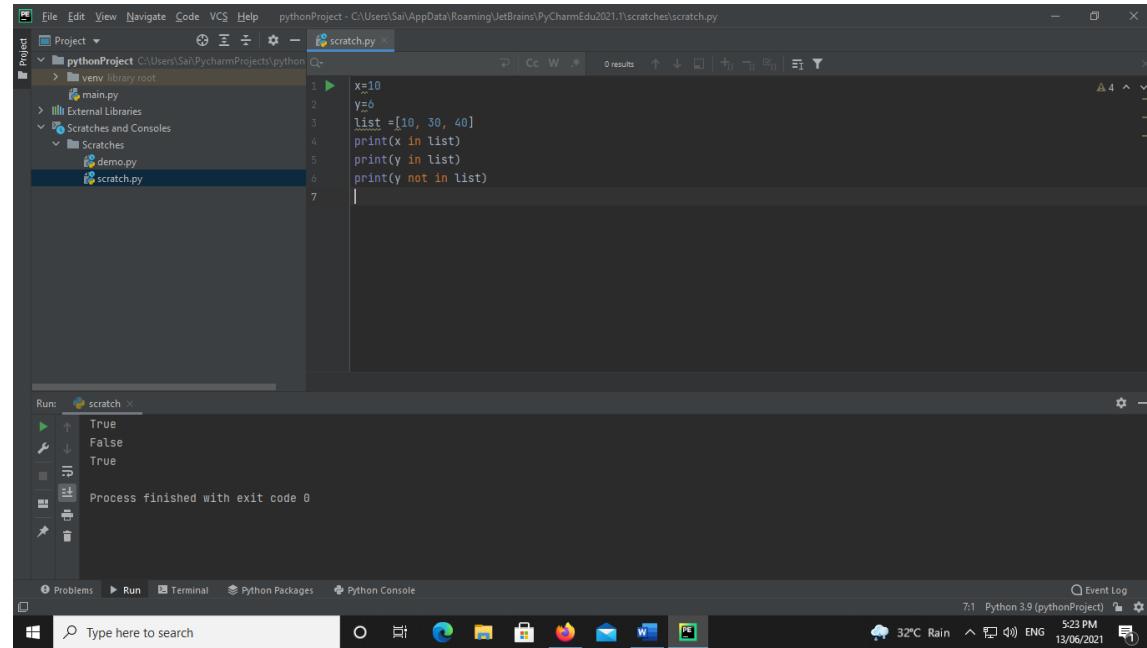
- **Membership Operators**

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:-



The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, VCS, Help, and a project path: pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py. The Project tool window on the left shows a file structure with a 'pythonProject' folder containing 'main.py' and 'Scratches' folder with 'demo.py' and 'scratch.py'. The 'scratch.py' file is selected. The main editor window contains the Python code:

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

. Below the editor is the 'Run' tool window, which has a dropdown menu set to 'scratch'. It displays three run results: True, False, and True. At the bottom of the Run window, it says 'Process finished with exit code 0'. The bottom status bar shows the Python version as '7.1 Python 3.9 (pythonProject)', the date and time as '13/06/2021 5:23 PM', and the system temperature as '32°C Rain'. The taskbar at the very bottom of the screen shows various application icons.

Task:-5

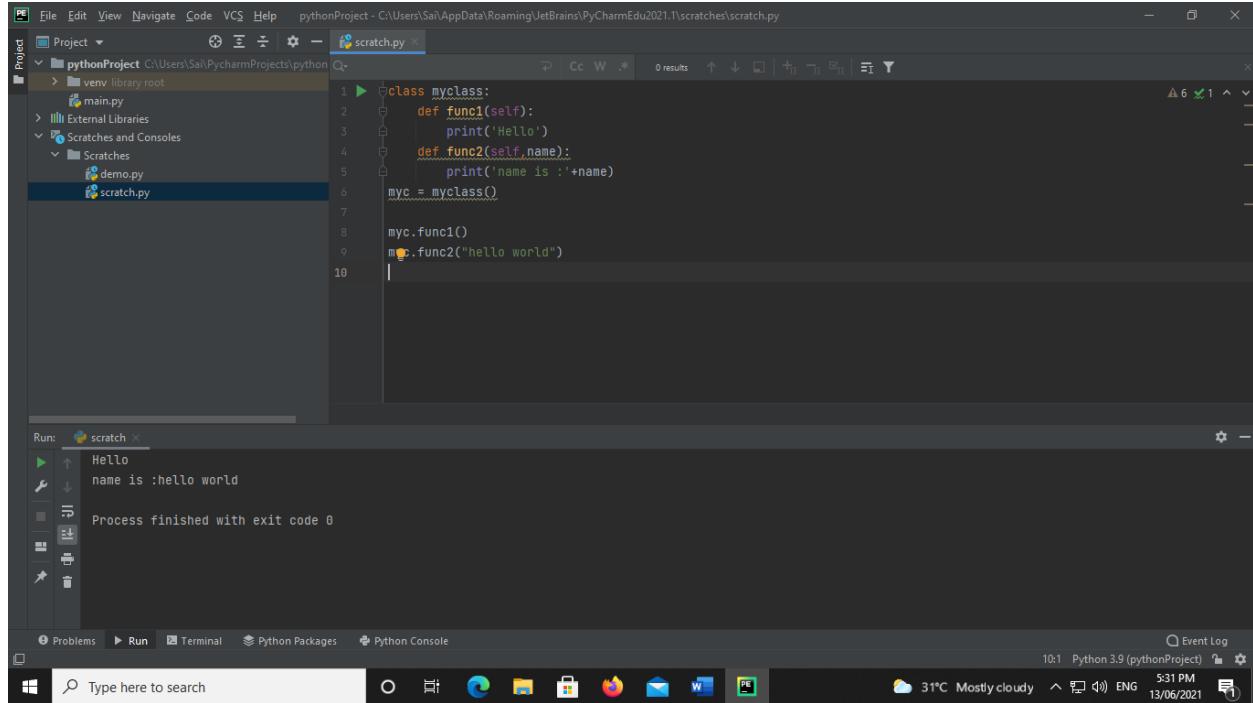
Class:

- 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:



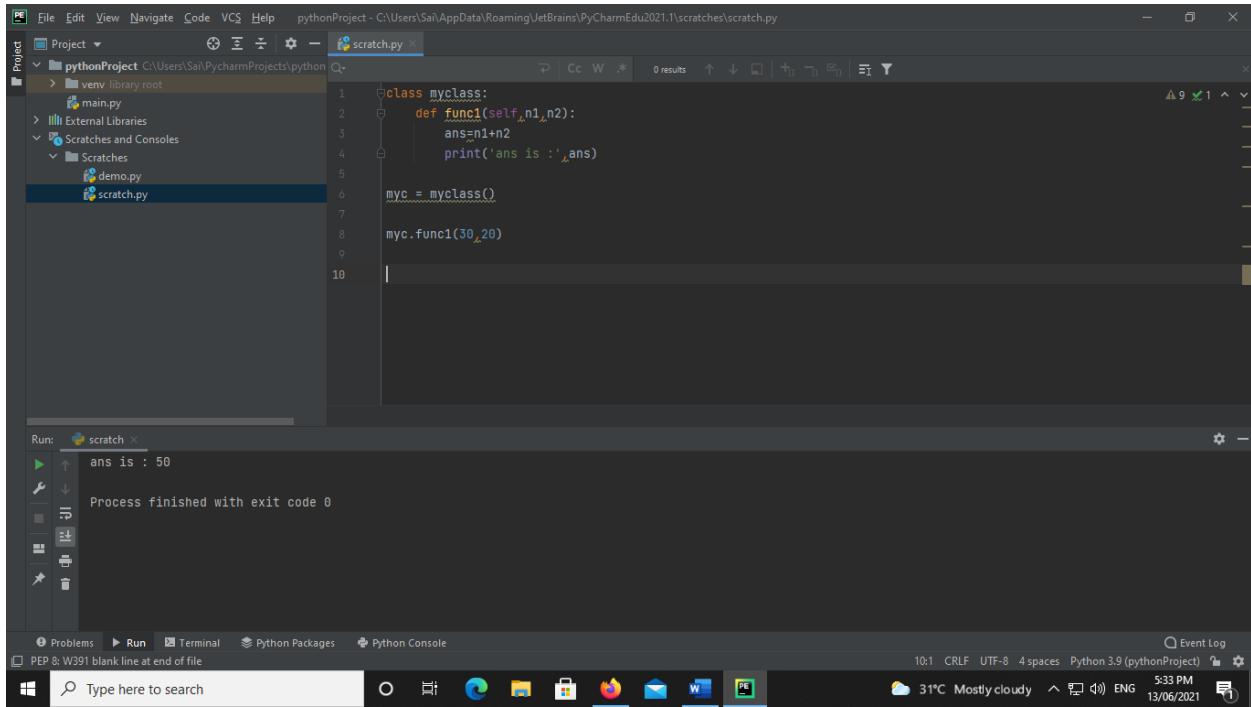
The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, VCS, Help, and pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py. The left sidebar displays the Project tree with a venv library root, main.py, External Libraries, Scratches and Consoles, and a Scratches folder containing demo.py and scratch.py. The main editor window contains the Python code provided above. Below the editor is the Run tool window, which shows the output of the script: "Hello" and "name is :hello world". At the bottom, the status bar indicates 10:1 Python 3.9 (pythonProject), 31°C Mostly cloudy, 5:31 PM, and 13/06/2021.

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:-



The screenshot shows the PyCharm IDE interface. The top bar displays the project path: "pythonProject C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py". The main area shows the Python code for a class named "myclass" with a method "func1". The code is run, and the output window shows the result: "ans is : 50". The status bar at the bottom right indicates the date and time: "13/06/2021 5:33 PM".

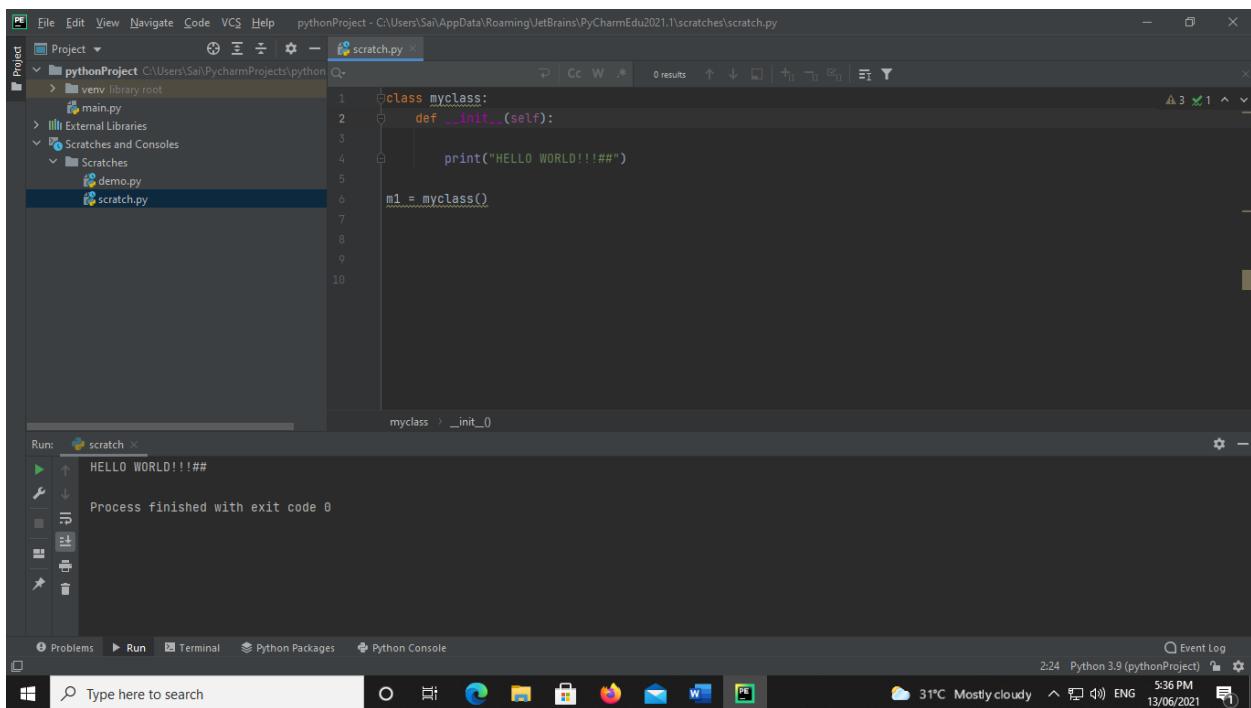
Python Constructors:-

- 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:-



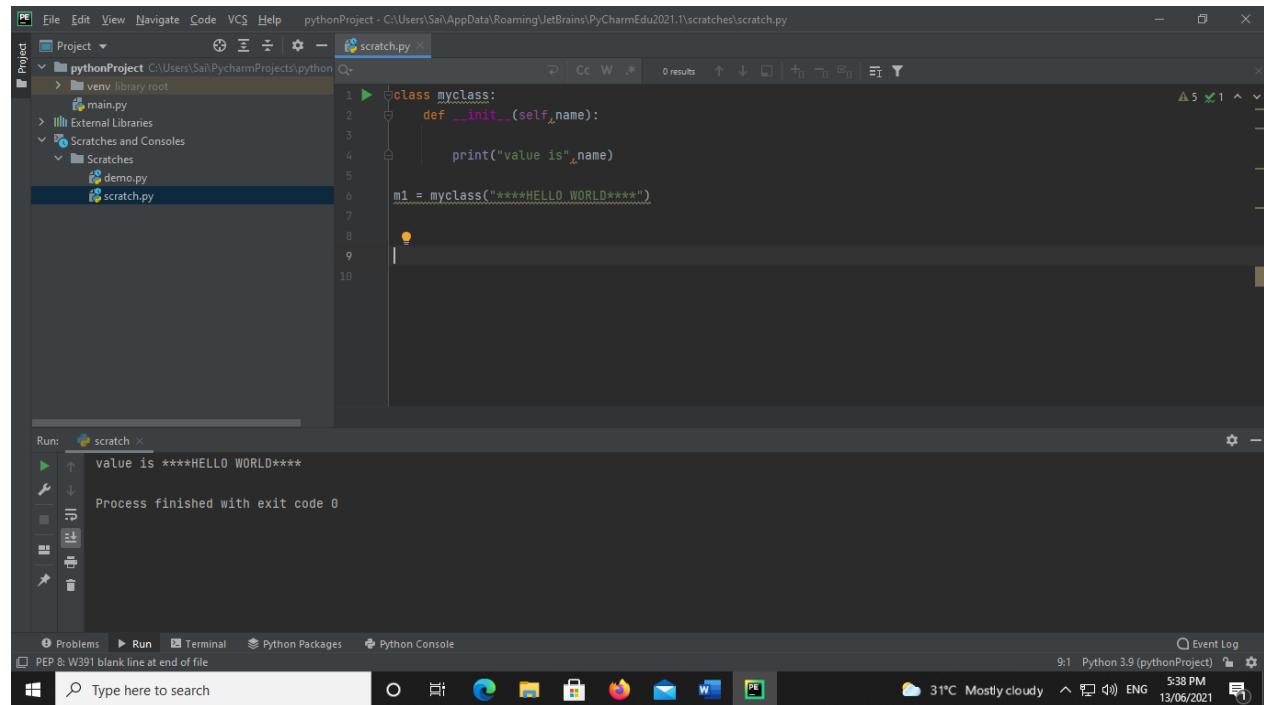
```
File Edit View Navigate Code VCS Help pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py  
Project pythonProject C:\Users\Sai\PycharmProjects\python Q... scratch.py  
File Edit View Navigate Code VCS Help pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py  
Project pythonProject C:\Users\Sai\PycharmProjects\python Q... scratch.py  
1 class myclass:  
2     def __init__(self):  
3         print("HELLO WORLD!!!##")  
4  
5 m1 = myclass()  
6  
7  
8  
9  
10  
myclass > __init__()  
Run: scratch  
▶ HELLO WORLD!!!##  
Process finished with exit code 0  
Problems Run Terminal Python Packages Python Console Event Log  
2:24 Python 3.9 (pythonProject) 5:36 PM 13/06/2021  
Windows Type here to search 31°C Mostly cloudy ENG 13/06/2021
```

- Parameterized constructor

Code:

```
class myclass:  
    def __init__(self,name):  
        print("value is",name)  
  
m1 = myclass("****HELLO WORLD****")
```

Output:-



The screenshot shows the PyCharm IDE interface. The project tree on the left shows a 'pythonProject' with files 'main.py', 'External Libraries', and 'Scratches' containing 'demo.py' and 'scratch.py'. The 'scratch.py' file is open in the editor, displaying the provided Python code. The 'Run' tool window at the bottom shows the output of running 'scratch.py': 'value is ****HELLO WORLD****' followed by 'Process finished with exit code 0'. The status bar at the bottom right indicates the environment is 'Python 3.9 (pythonProject)'.

Assign string value to class variable using Method

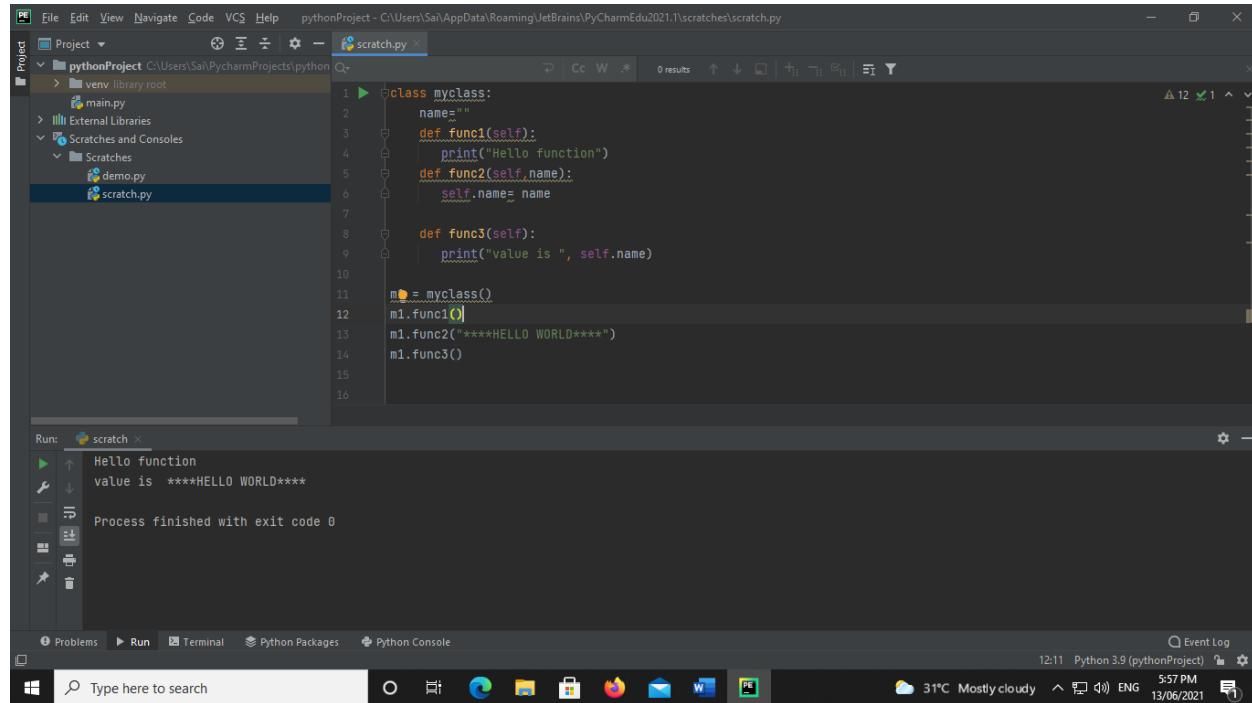
Code:

```
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:-



The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject
- File:** scratch.py
- Code:** The Python code defined in the question is displayed.
- Run:** The output of the code execution is shown in the Run tab:
 - Line 1: Hello function
 - Line 2: value is *****HELLO WORLD*****
 - Line 3: Process finished with exit code 0
- Bottom Status Bar:** Shows the date (13/06/2021), time (5:57 PM), and weather (31°C, Mostly cloudy).

Assign string value to class variable using Constructor

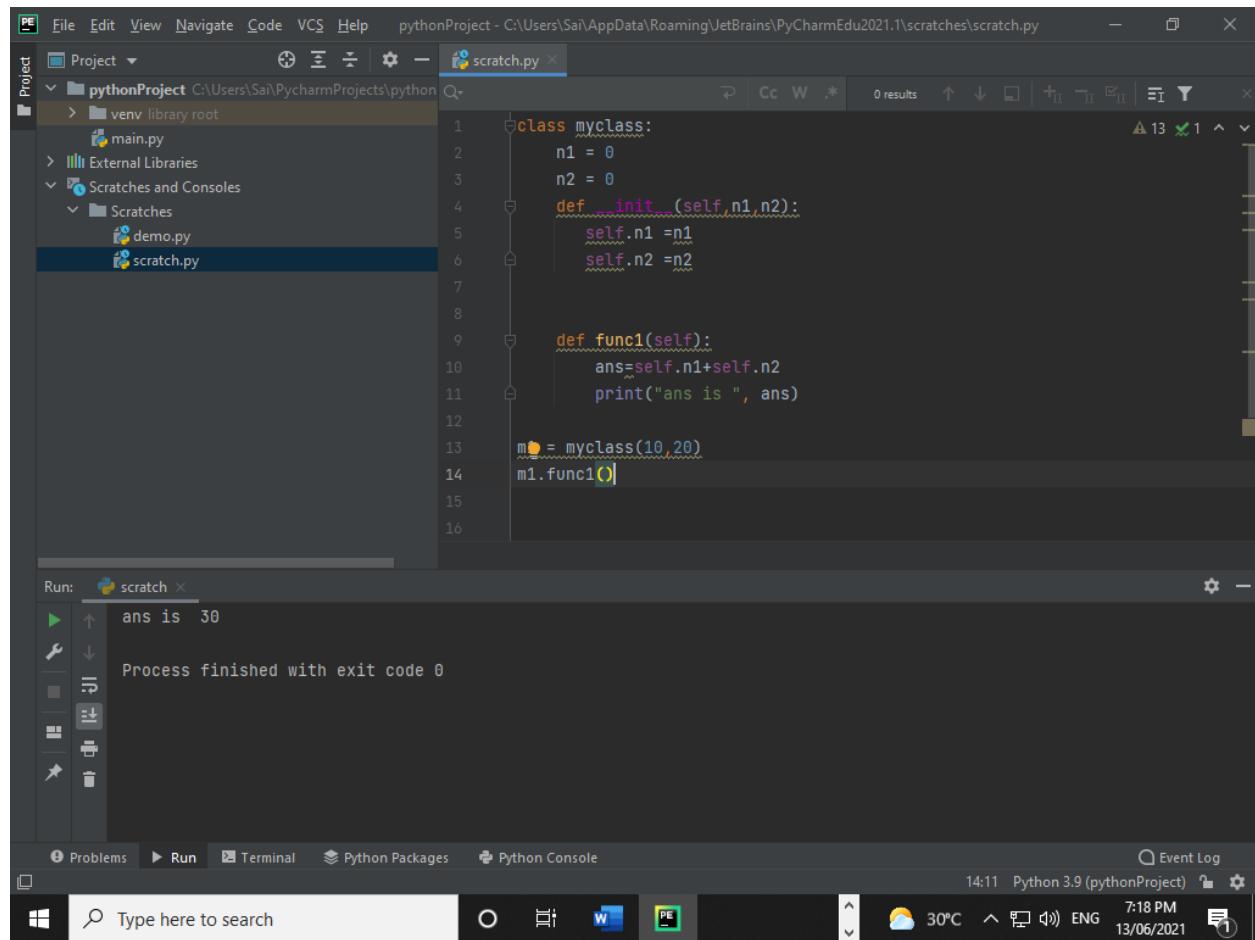
Code:

```
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:-



The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject
- Scratches:** A scratch file named "scratch.py" is open in the editor.
- Code:** The code defines a class `myclass` with two integer attributes `n1` and `n2`, and two methods `__init__` and `func1`. The `func1` method prints the sum of `n1` and `n2`.
- Run Tab:** The output window shows the execution results:
 - Line 13: `m1 = myclass(10,20)`
 - Line 14: `m1.func1()`
 - Output: `ans is 30`
 - Status: `Process finished with exit code 0`
- Bottom Status Bar:** Shows the time as 14:11, Python version as Python 3.9 (pythonProject), and system information including weather (30°C), language (ENG), and date (13/06/2021).

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

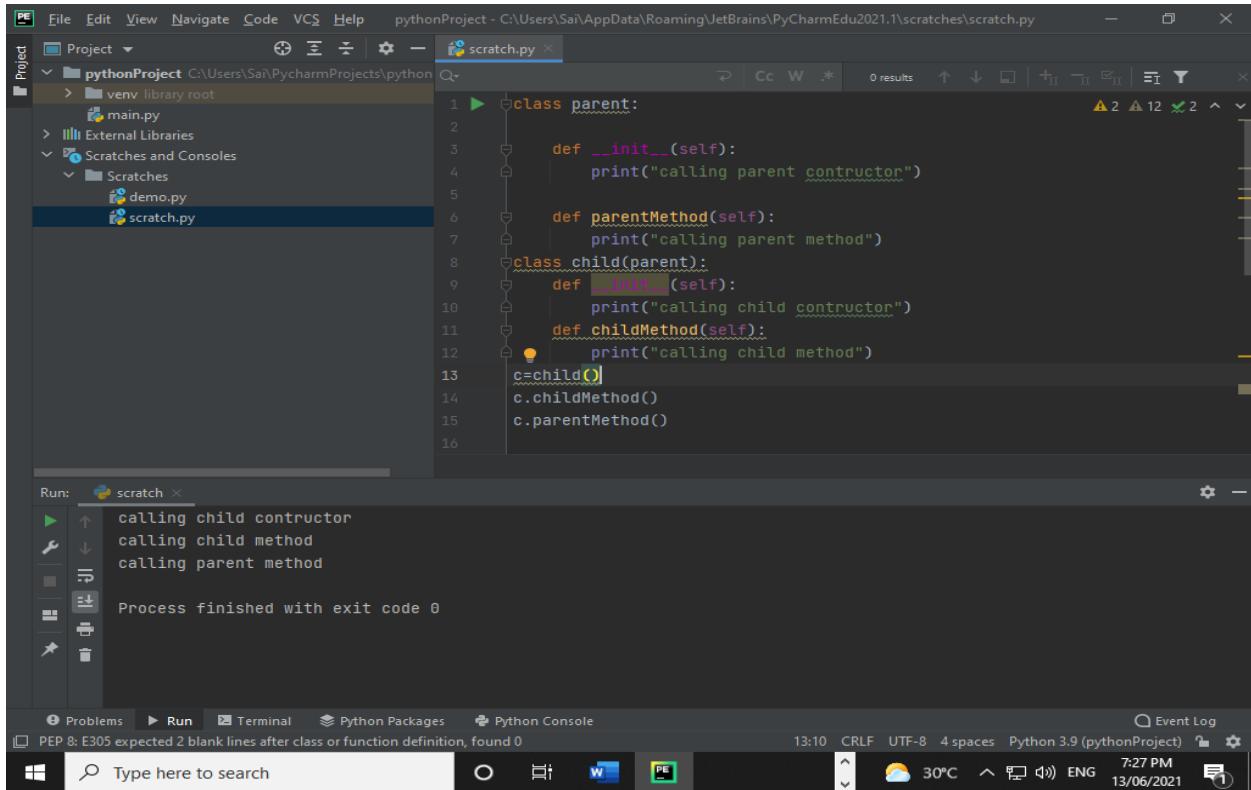
Code:

```
• class parent:

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

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

Output:-



The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject C:\Users\SAI\PycharmProjects\python
- File:** scratch.py
- Code:**

```
1 class parent:
2     def __init__(self):
3         print("calling parent constructor")
4     def parentMethod(self):
5         print("calling parent method")
6 class child(parent):
7     def __init__(self):
8         print("calling child constructor")
9     def childMethod(self):
10        print("calling child method")
11 c=child()
12 c.childMethod()
13 c.parentMethod()
```
- Run:** scratch
- Output:**

```
calling child constructor
calling child method
calling parent method
```
- Bottom Status Bar:** PEP 8: E305 expected 2 blank lines after class or function definition, found 0
- Bottom Right:** Event Log, 13:10, CRLF, UTF-8, 4 spaces, Python 3.9 (pythonProject), 30°C, ENG, 7:27 PM, 13/06/2021

- Multi-level Inheritance

Code:

```
class parent:  
    def __init__(self):  
        print("calling parent constructor")  
    def parentMethod(self):  
        print("calling parent method")  
class child(parent):  
    def __init__(self):  
        print("calling child constructor")  
    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:-

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject
- Scratches:** scratch.py
- Code:** The code defines three classes: parent, child, and Subchild, demonstrating multi-level inheritance. The output of the code execution is displayed in the Run tab.
- Output:**

```
calling sub child contrutor  
calling sub child method  
calling child method  
calling parent method
```
- Bottom Status Bar:** PEP 8: E302 expected 2 blank lines, found 0
- System Status:** 7:1 Python 3.9 (pythonProject) 7:35 PM 13/06/2021

- 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:

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py
- Code Editor:** The file scratch.py contains the provided Python code for multiple inheritance.
- Run Tab:** The "Run" tab shows the output of the code execution:
 - parent1 method
 - parent2 method
 - child method
- Status Bar:** The status bar at the bottom right shows the date and time (13/06/2021, 7:47 PM), weather (30°C), and system information (ENG).

- Hierarchical Inheritance

Code:

```
class parent:  
    def __init__(self):  
        print("calling parent constructor")  
    def parentMethod(self):  
        print("calling parent method")  
class child(parent):  
    def __init__(self):  
        print("calling child constructor")  
    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:-

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject
- Scratches:** scratch.py
- Code Content:**

```
1  > class parent:  
2      def __init__(self):  
3          print("calling parent constructor")  
4      def parentMethod(self):  
5          print("calling parent method")  
6  class child(parent):  
7      def __init__(self):  
8          print("calling child constructor")  
9      def childMethod(self):  
10         print("calling child method")  
11  class child2(parent):  
12      def __init__(self):  
13          print("calling child2 contrutor")  
14      def childMethod2(self):  
15          print("calling child2 method")  
16  sc=child2()  
17  sc.childMethod2()  
18  sc.parentMethod()
```
- Run Output:**

```
calling child2 contrutor  
calling child2 method  
calling parent method
```
- Status Bar:** CRLF, UTF-8, 4 spaces, Python 3.9 (pythonProject), 7:52 PM, 13/06/2021

- 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:-

The screenshot shows the PyCharm IDE interface with the following details:

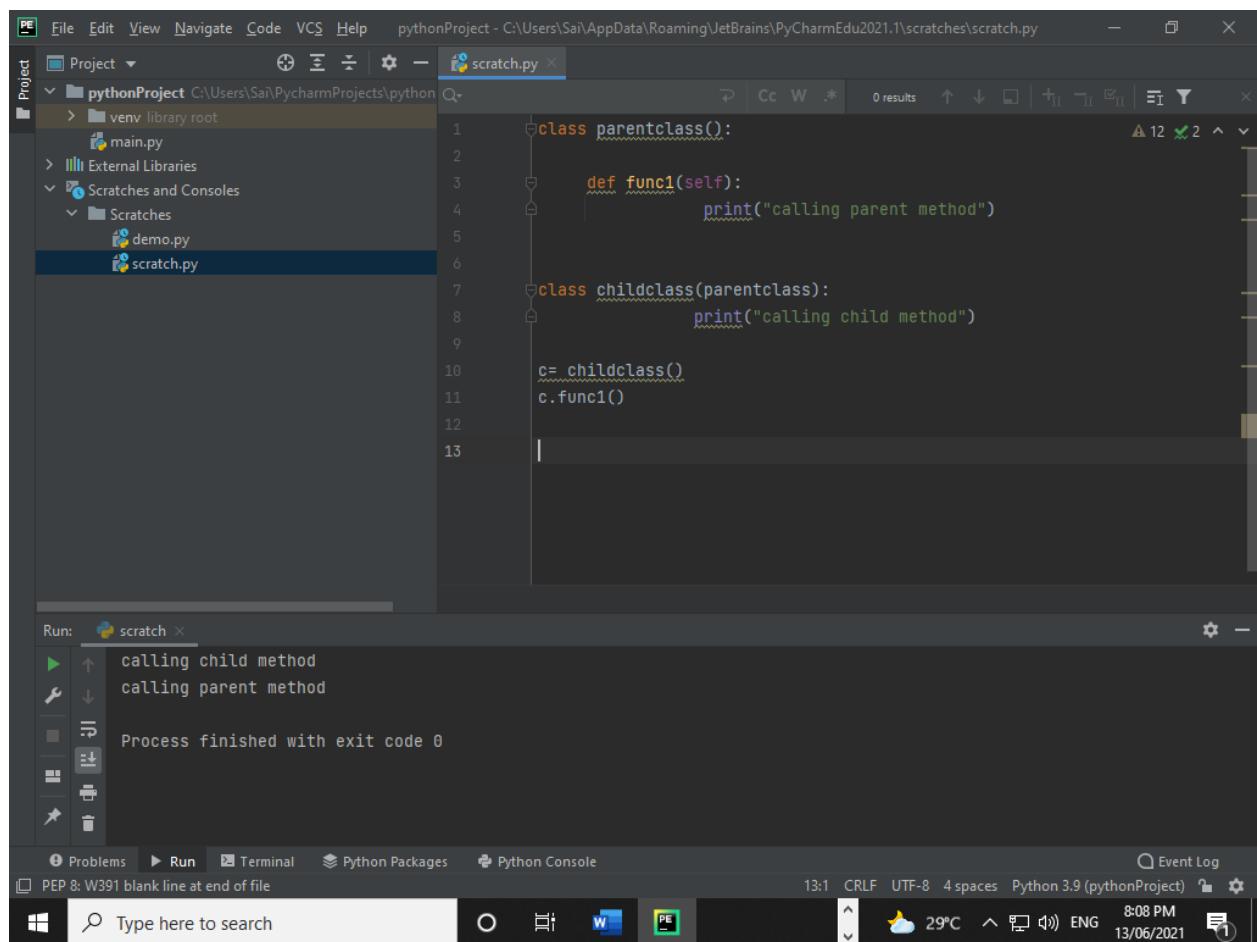
- Project:** pythonProject
- Scratches:** scratch.py
- Code:** The code is identical to the one provided in the "Code:" section.
- Run:** The run configuration is set to "scratch".
- Output:** The terminal shows the following output:

```
child method
child method2
parent1 method
Process finished with exit code 0
```
- Environment:** Python 3.9 (pythonProject)
- Timestamp:** 19:19 13/06/2021

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:



The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py
- Code Editor:** scratch.py

```
PE File Edit View Navigate Code VCS Help pythonProject - C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py — ×
Project pythonProject C:\Users\Sai\AppData\Roaming\JetBrains\PyCharmEdu2021.1\scratches\scratch.py
pythonProject > venv library root
  main.py
> External Libraries
Scatches and Consoles
  Scratches
    demo.py
    scratch.py

scratches.py
1 class parentclass():
2
3     def func1(self):
4         print("calling parent method")
5
6
7 class childclass(parentclass):
8
9     def func1(self):
10        print("calling child method")
11
12
13 c= childclass()
c.func1()
```

- Run:** scratch

```
Process finished with exit code 0
```

- Bottom Bar:** Problems, Run, Terminal, Python Packages, Python Console, Event Log.
- System Tray:** PEP 8: W391 blank line at end of file, 13:1, CRLF, UTF-8, 4 spaces, Python 3.9 (pythonProject), 8:08 PM, 13/06/2021, 29°C, ENG.

- Overriding Method

Code:

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** pythonProject
- File:** scratch.py
- Code Content:**

```
1 class myclass():
2     def sum(self, n1, n2):
3         ans = n1+n2
4         print("ans:", ans)
5
6     def sum(self, n1, n2, n3):
7         ans = n1+n2+n3
8         print("ans:", ans)
9
10    p = myclass()
11    p.sum(30, 20, 10)
```

- Run Output:**

```
ans: 60
Process finished with exit code 0
```

- Bottom Status Bar:** PEP 8: E231 missing whitespace after ','
- System Tray:** Weather (29°C), Language (ENG), Date (13/06/2021), Time (8:15 PM)

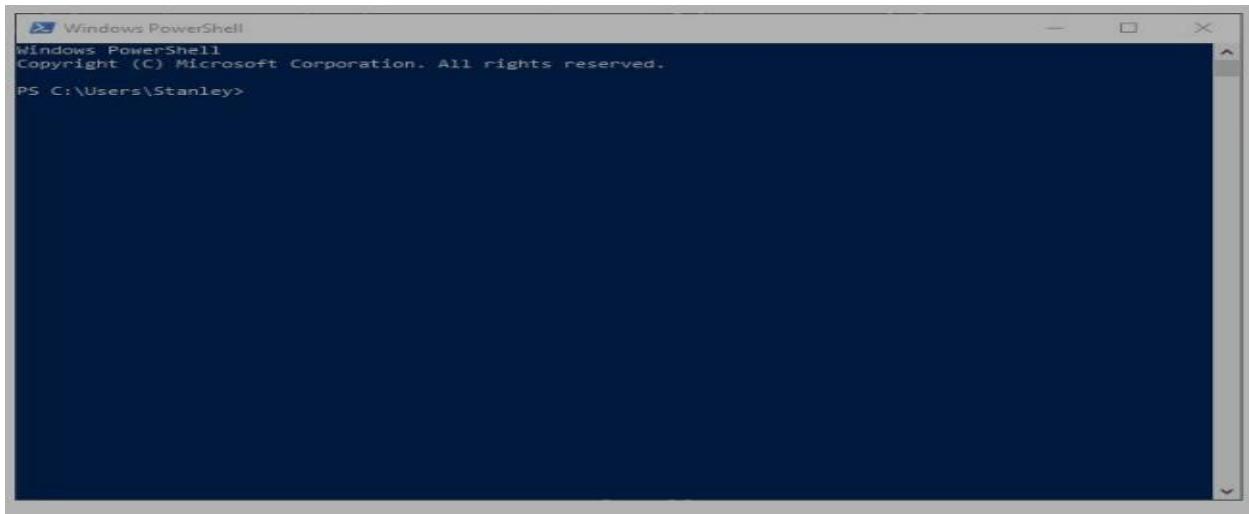
Task:6

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
```

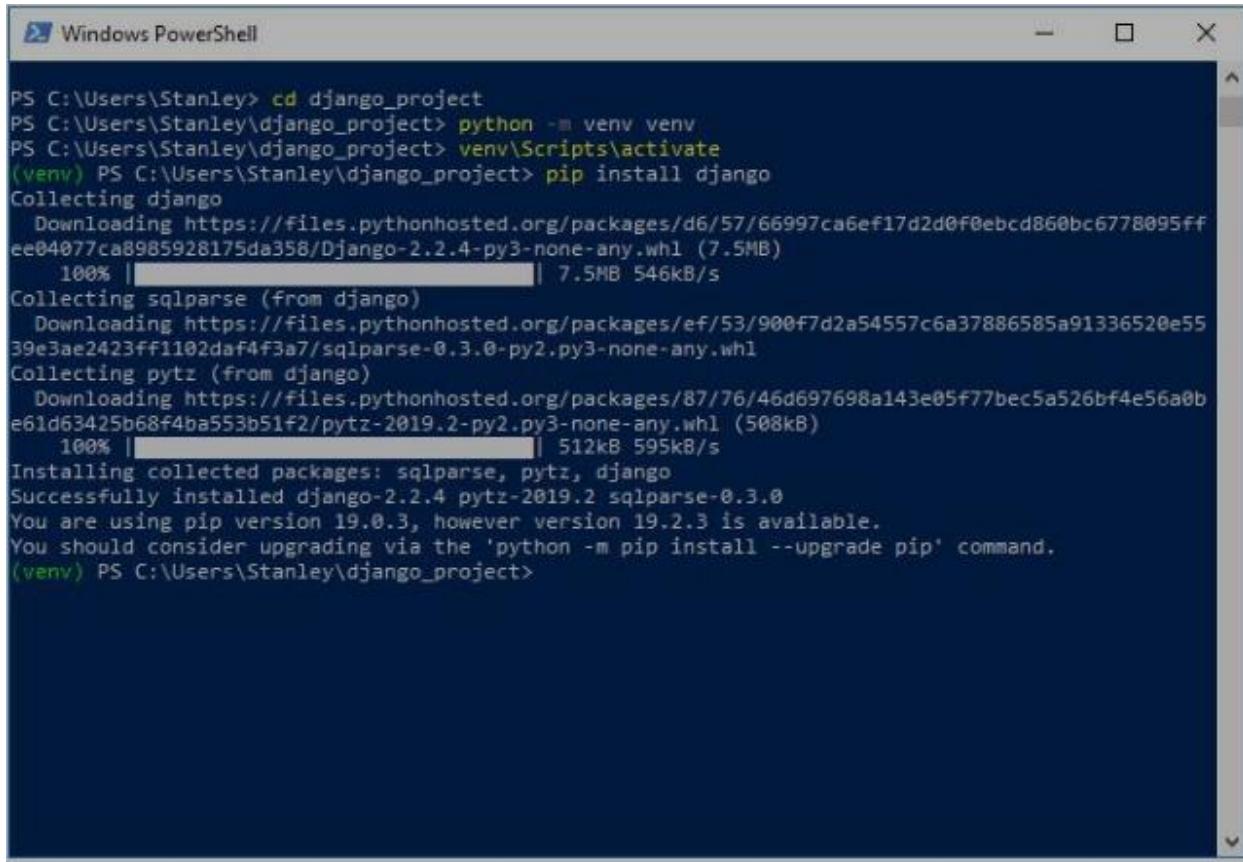
Step 6-Activate virtual environment

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

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

Step 7-install Django

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



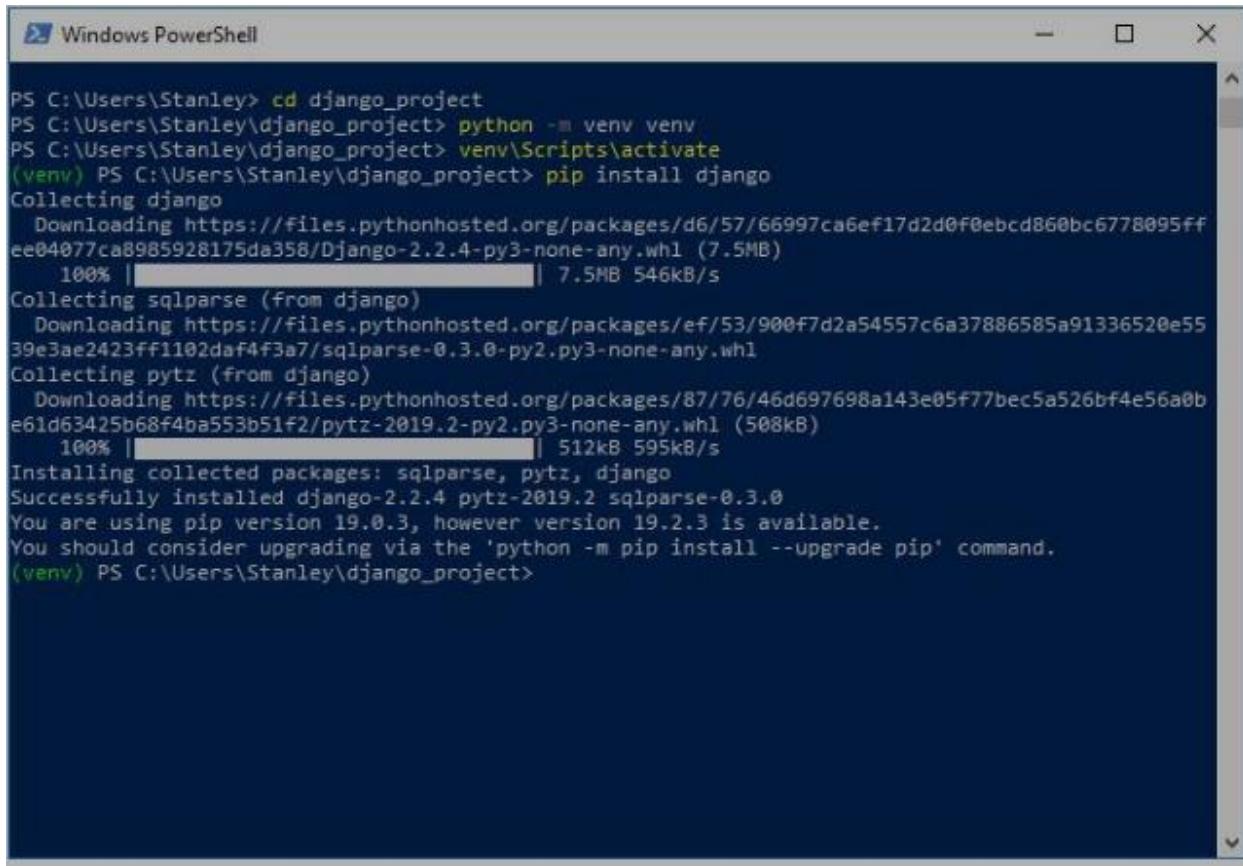
```
PS C:\Users\Stanley> cd django_project
PS C:\Users\Stanley\django_project> python -m venv venv
PS C:\Users\Stanley\django_project> venv\Scripts\activate
(venv) PS C:\Users\Stanley\django_project> pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/d6/57/66997ca6ef17d2d0f0ebcd860bc6778095ff
ee04077ca8985928175da358/Django-2.2.4-py3-none-any.whl (7.5MB)
    100% |██████████| 7.5MB 546kB/s
Collecting sqlparse (from django)
  Downloading https://files.pythonhosted.org/packages/ef/53/900f7d2a54557c6a37886585a91336520e55
39e3ae2423ff1102daf4f3a7/sqlparse-0.3.0-py2.py3-none-any.whl
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/87/76/46d697698a143e05f77bec5a526bf4e56a0b
e61d63425b68f4ba553b51f2/pytz-2019.2-py2.py3-none-any.whl (508kB)
    100% |██████████| 512kB 595kB/s
Installing collected packages: sqlparse, pytz, django
Successfully installed django-2.2.4 pytz-2019.2 sqlparse-0.3.0
You are using pip version 19.0.3, however version 19.2.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
(venv) PS C:\Users\Stanley\django_project>
```

Step 8-Start new Project

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

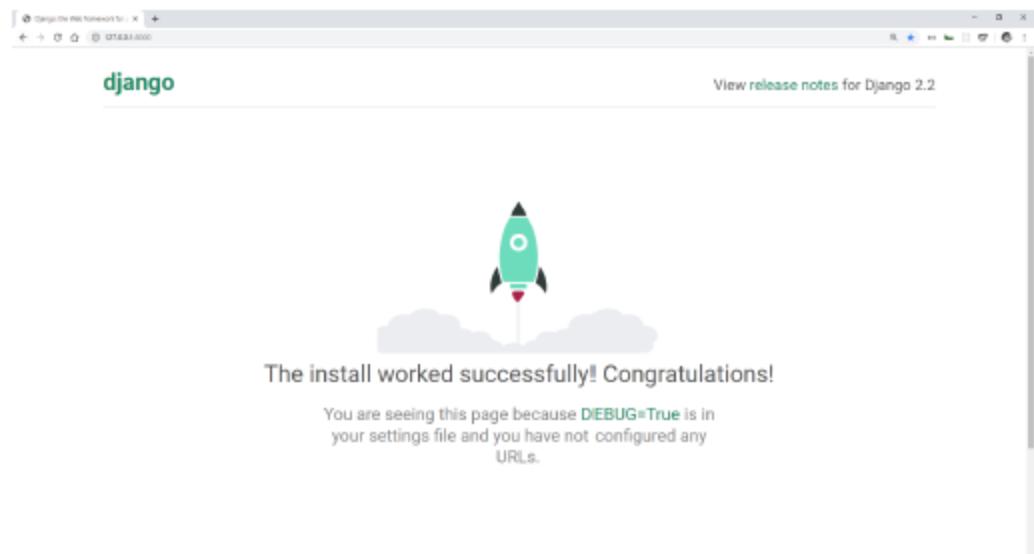
```
(venv)> cd testsite
```

Step 9-Run the server



```
PS C:\Users\Stanley> cd django_project
PS C:\Users\Stanley\django_project> python -m venv venv
PS C:\Users\Stanley\django_project> venv\Scripts\activate
(venv) PS C:\Users\Stanley\django_project> pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/d6/57/66997ca6ef17d2d0f0ebcd860bc6778095ff
ee04077ca8985928175da358/Django-2.2.4-py3-none-any.whl (7.5MB)
    100% |██████████| 7.5MB 546kB/s
Collecting sqlparse (from django)
  Downloading https://files.pythonhosted.org/packages/ef/53/900f7d2a54557c6a37886585a91336520e55
39e3ae2423ff1102daf4f3a7/sqlparse-0.3.0-py2.py3-none-any.whl
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/87/76/46d697698a143e05f77bec5a526bf4e56a0b
e61d63425b68f4ba553b51f2/pytz-2019.2-py2.py3-none-any.whl (508kB)
    100% |██████████| 512kB 595kB/s
Installing collected packages: sqlparse, pytz, django
Successfully installed django-2.2.4 pytz-2019.2 sqlparse-0.3.0
You are using pip version 19.0.3, however version 19.2.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
(venv) PS C:\Users\Stanley\django_project>
```

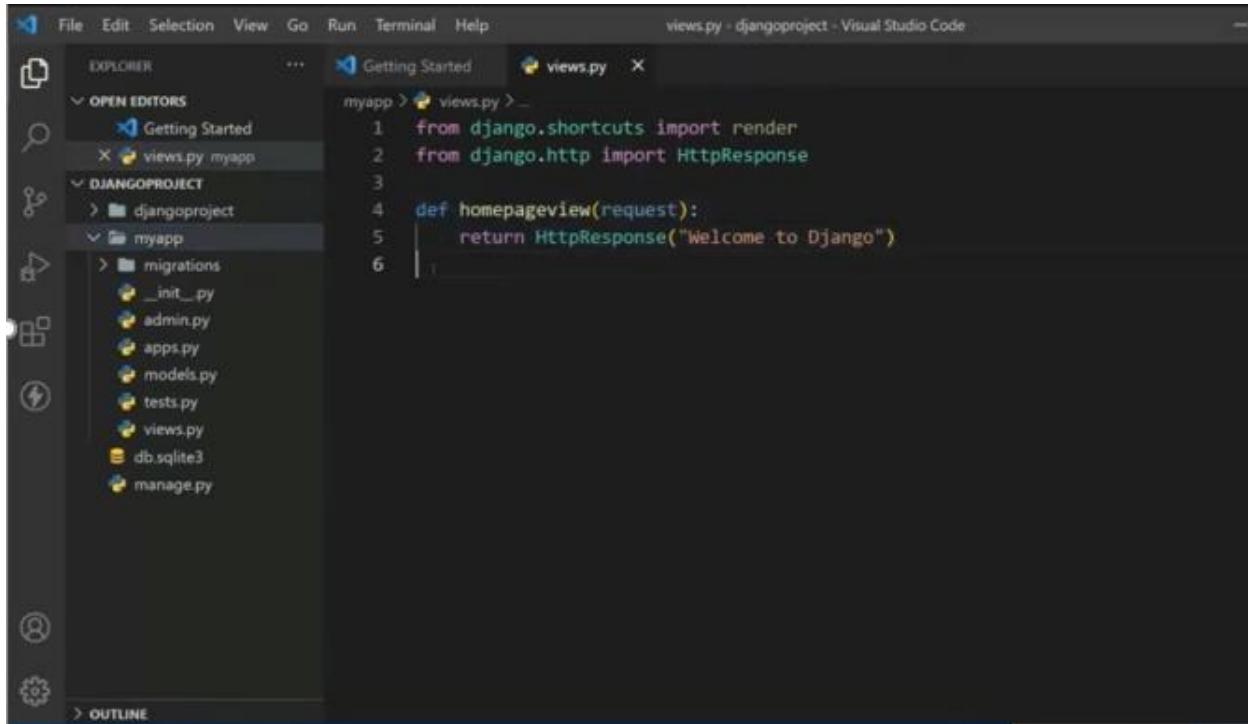
Welcome Screen



Task:7

Print Welcome to Django.

views.py



The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor Bar:** views.py - djangoproject - Visual Studio Code.
- Explorer:** Shows the project structure:
 - OPEN EDITORS: Getting Started, views.py (myapp)
 - DJANGOPROJECT: djangoproject (expanded), myapp (expanded)
 - migrations: __init__.py, admin.py, apps.py, models.py, tests.py
 - views.py
 - db.sqlite3
 - manage.py
- Code Editor:** The views.py file contains the following code:

```
from django.shortcuts import render
from django.http import HttpResponseRedirect

def homepageview(request):
    return HttpResponseRedirect("Welcome to Django")
```
- Bottom Bar:** OUTLINE.

Urls.py my app

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Open Editors:** Getting Started, views.py - myapp, urls.py - myapp (highlighted), urls.py - djangoproject.
- Explorer:** Shows the project structure:
 - DJANGOPROJECT: __init__.py, asgi.py, settings.py, urls.py, wsgi.py
 - myapp: migrations, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, views.py, db.sqlite3, manage.py
- Terminal:** Getting Started, views.py, urls.py - myapp, urls.py - djangoproject.
- Code Editor:** urls.py - myapp (containing code)

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.homepageview, name='home'),
```

Url.py

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Open Editors:** Getting Started, views.py - myapp, urls.py - myapp, settings.py - djangoproject (highlighted), urls.py - djangoproject.
- Explorer:** Shows the project structure:
 - DJANGOPROJECT: __init__.py, asgi.py, settings.py, urls.py, wsgi.py
 - djangoproject: __pycache__, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, wsgi.py
 - myapp: migrations, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py
- Terminal:** Getting Started, views.py, urls.py - myapp, settings.py - djangoproject, urls.py - djangoproject.
- Code Editor:** urls.py - djangoproject (containing code)

```
Function views
1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')

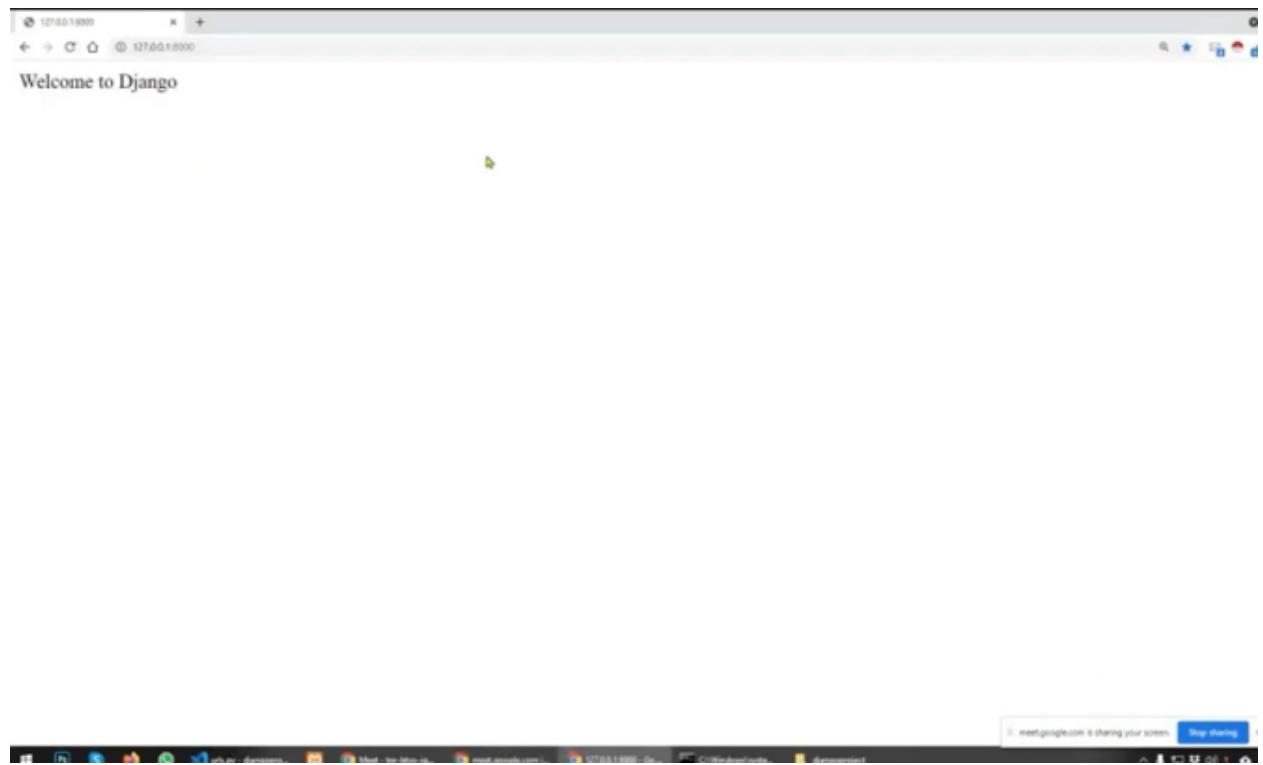
Class-based views
1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')

Including another URLconf
1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('myapp.urls')),
```

Output:



Task:8

In Website How to create pages in Django

The screenshot shows the Visual Studio Code interface with a Django project structure. The Explorer sidebar on the left lists files and folders: urls.py (djangoproject), urls.py (myapp), views.py (myapp), DJANGOPROJECT (djangoproject), myapp (containing __pycache__, migrations, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, views.py), templates (containing about.html, contact.html, home.html), db.sqlite3, and manage.py. The Outline sidebar shows the current file being edited: views.py (myapp). The main editor area displays Python code for views:

```
from django.shortcuts import render
from django.http import HttpResponseRedirect
def homepageview(request):
    return render(request,'home.html')
def aboutpageview(request):
    return render(request,'about.html')
def contactpageview(request):
    return render(request,'contact.html')
```

The terminal at the bottom shows the development server output:

```
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[03/Jun/2021 16:08:33] "GET / HTTP/1.1" 200 26
Not Found: /favicon.ico
[03/Jun/2021 16:08:33] "GET /favicon.ico HTTP/1.1" 404 2613
[03/Jun/2021 16:08:41] "GET /about HTTP/1.1" 200 13
[03/Jun/2021 16:08:44] "GET /contact HTTP/1.1" 200 15
D:\djangoproject\myapp\views.py changed, reloading.
```

Navigation bar

The screenshot shows the Visual Studio Code interface with a Django template file open. The Explorer sidebar lists files: urls.py (djangoproject), urls.py (myapp), views.py (myapp), settings.py (djangoproject), and home.html (templates). The Outline sidebar shows the current file being edited: home.html (templates). The main editor area displays HTML code:

```
<h1>Home</h1>
<a href="/">Home</a>
<a href="/about">About</a>
<a href="/contact">Contact</a>
```

Output:-



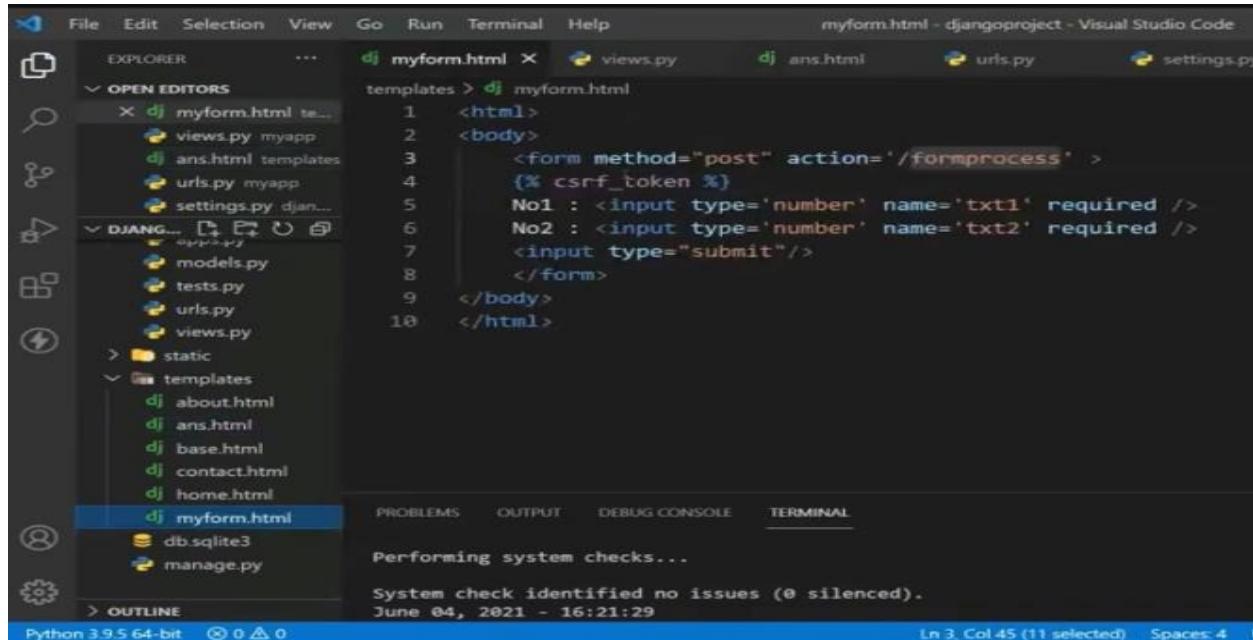
Home

[Home](#) [About](#) [Contact](#)

Task:9

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

Myform.html

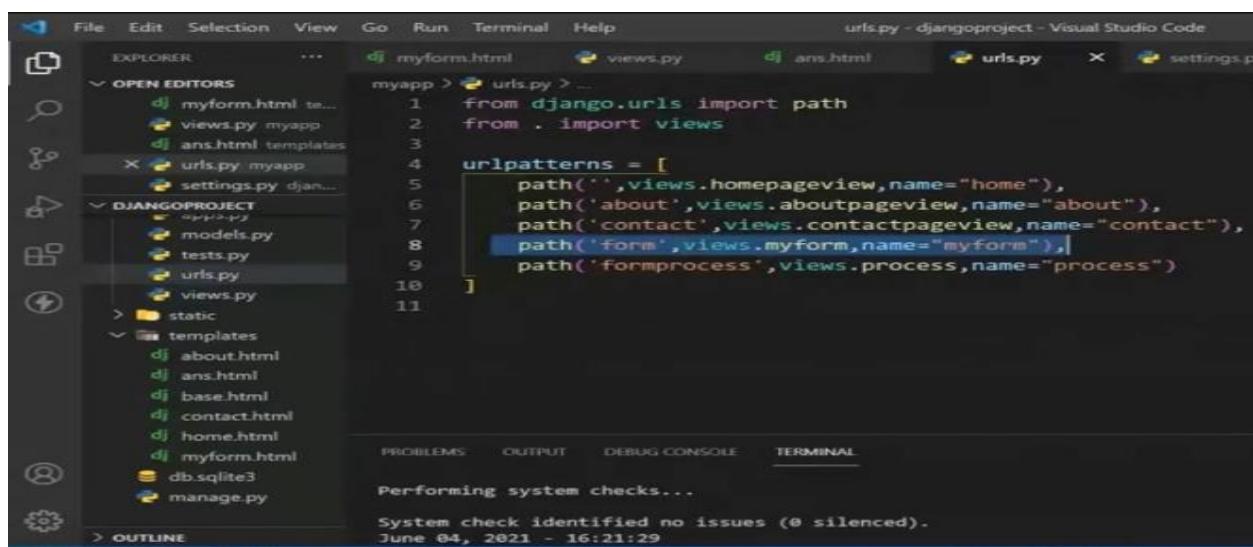


The screenshot shows the Visual Studio Code interface with the 'myform.html' file open in the editor. The file contains the following HTML code:

```
<html>
<body>
    <form method="post" action='/formprocess'>
        {% csrf_token %}
        No1 : <input type='number' name='txt1' required />
        No2 : <input type='number' name='txt2' required />
        </form>
    </body>
</html>
```

The 'templates' folder under 'myapp' is expanded in the Explorer sidebar. The status bar at the bottom indicates Python 3.9.5 64-bit, 0 errors, and 4 spaces.

Urls.py



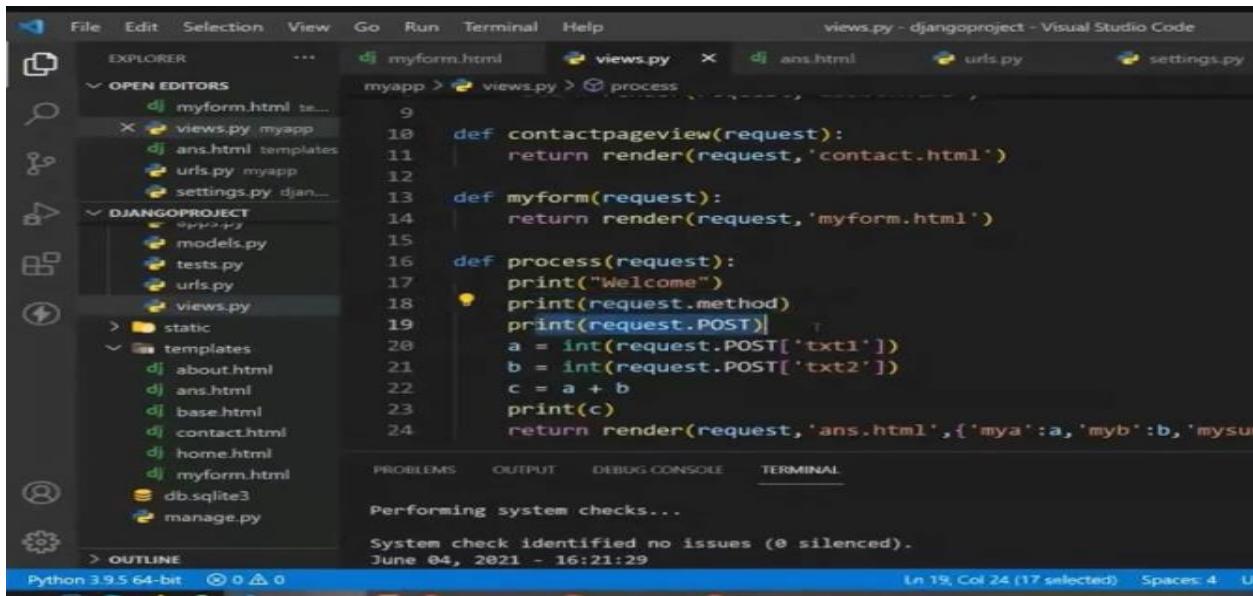
The screenshot shows the Visual Studio Code interface with the 'urls.py' file open in the editor. The file contains the following URL patterns:

```
from django.urls import path
from . import views

urlpatterns = [
    path('',views.homepageview,name="home"),
    path('about',views.aboutpageview,name="about"),
    path('contact',views.contactpageview,name="contact"),
    path('form',views.myform,name="myform"),
    path('formprocess',views.process,name="process")
]
```

The 'myapp' folder under 'myapp' is selected in the Explorer sidebar. The status bar at the bottom indicates Python 3.9.5 64-bit, 0 errors, and 4 spaces.

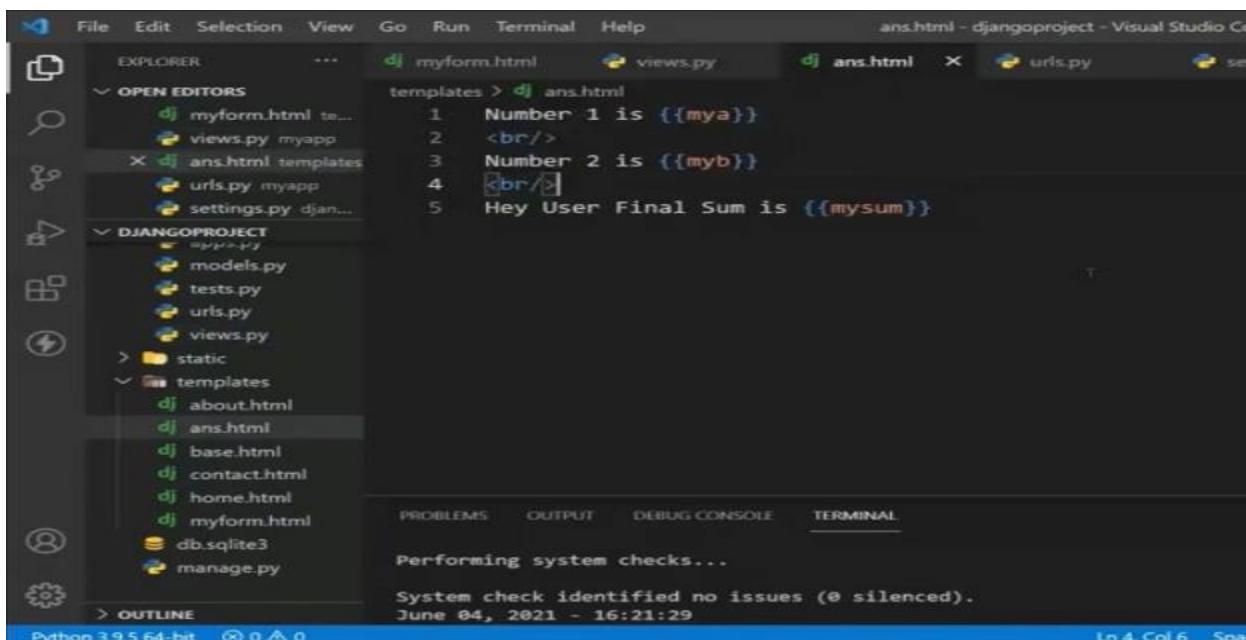
Views.py



The screenshot shows the Visual Studio Code interface with the 'views.py' file open in the editor. The code defines three functions: contactpageview, myform, and process. The process function prints 'Welcome', checks the request method, prints POST data, and calculates the sum of two integers from the POST data. The code is annotated with comments explaining its purpose.

```
9
10 def contactpageview(request):
11     return render(request,'contact.html')
12
13 def myform(request):
14     return render(request,'myform.html')
15
16 def process(request):
17     print("Welcome")
18     print(request.method)
19     print(request.POST)
20     a = int(request.POST['txt1'])
21     b = int(request.POST['txt2'])
22     c = a + b
23     print(c)
24     return render(request,'ans.html',{'mya':a,'myb':b,'mysu'}
```

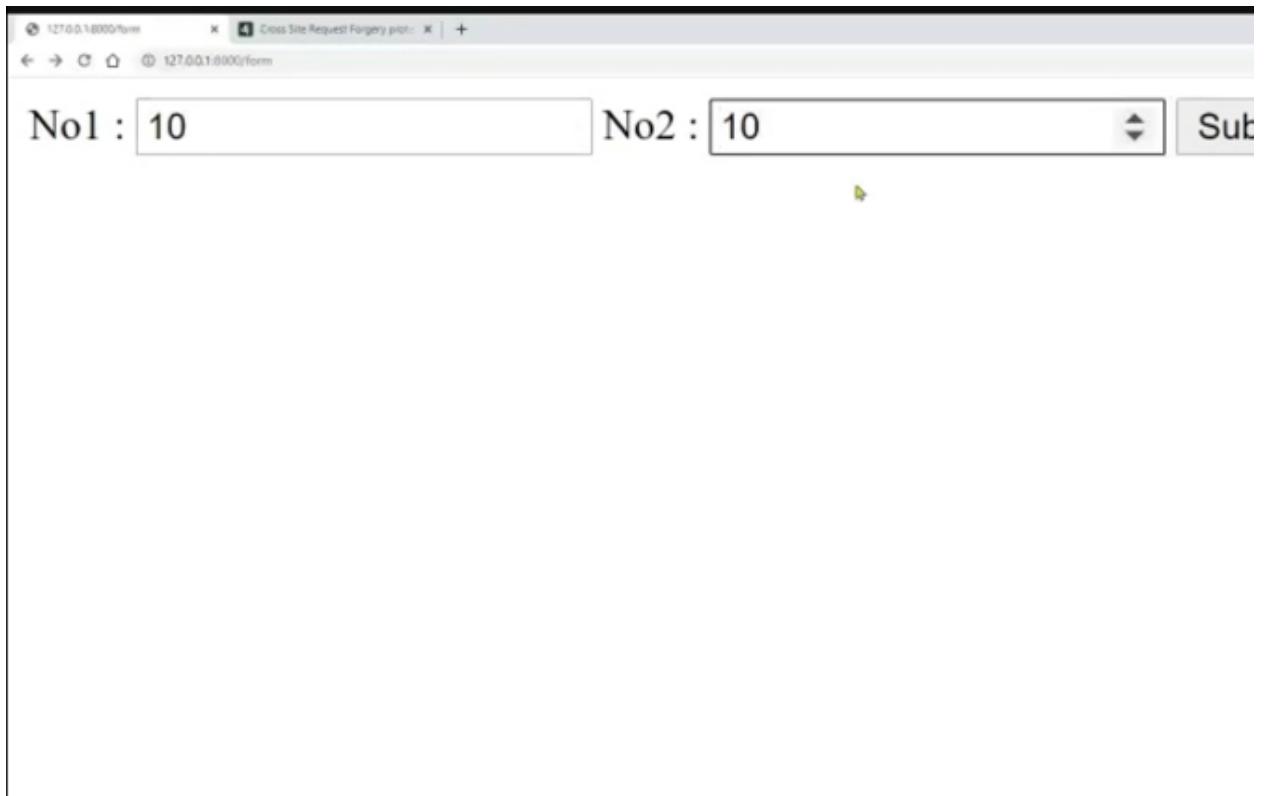
Ans.html



The screenshot shows the Visual Studio Code interface with the 'ans.html' template file open in the editor. The template contains five lines of HTML code that output variables: mya, myb, and mysum. The code is annotated with comments explaining its purpose.

```
1 Number 1 is {{mya}}
2 <br/>
3 Number 2 is {{myb}}
4 <br/>
5 Hey User Final Sum is {{mysum}}
```

Output:-



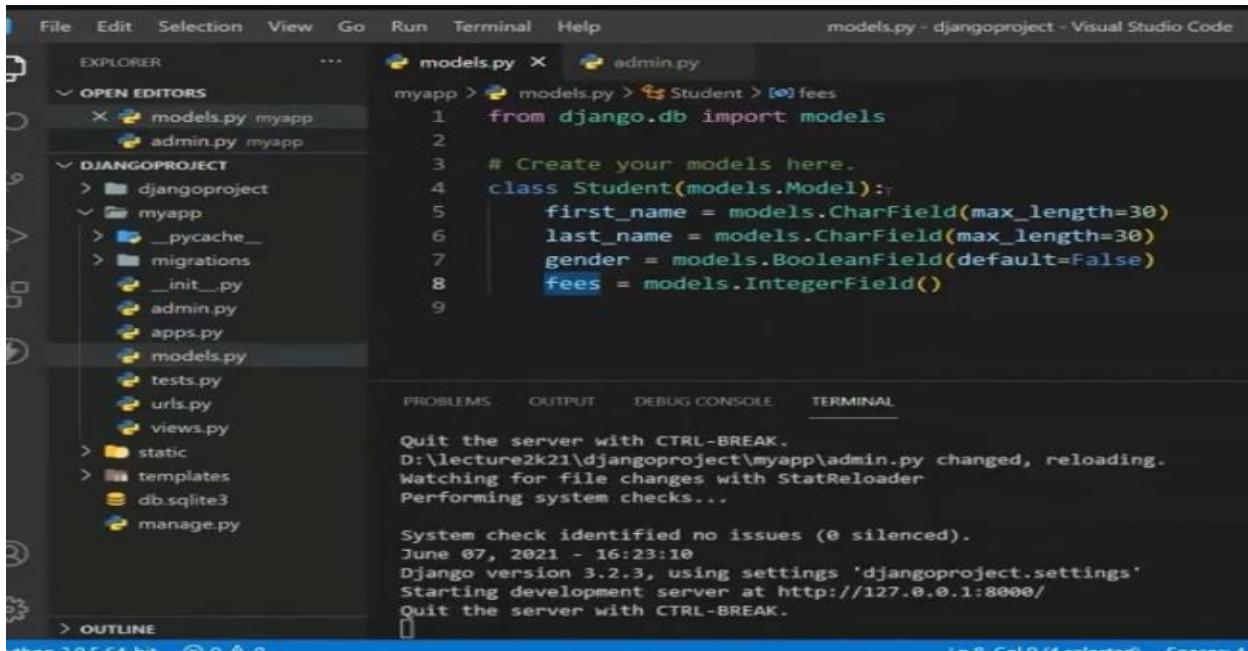
A screenshot of a web browser window titled "Cross Site Request Forgery protection". The address bar shows the URL "127.0.0.1:8000/form". The page contains a form with two input fields and a submit button. The first input field is labeled "No1 : [10]" and the second is labeled "No2 : [10]". To the right of the second input field is a dropdown arrow. To the right of the dropdown arrow is a submit button labeled "Submit".



Task:10

How to Use Database in Django.

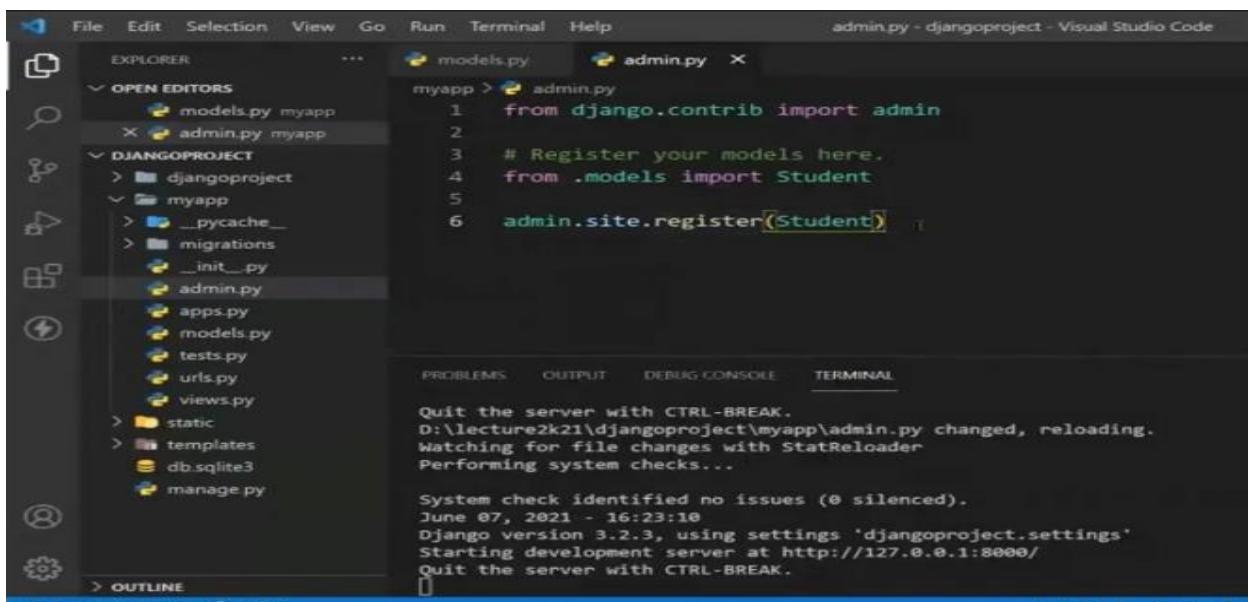
Model.py file



The screenshot shows the Visual Studio Code interface with the 'models.py' file open in the editor. The code defines a 'Student' model with fields for first name, last name, gender, and fees.

```
models.py - djangoproject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER OPEN EDITORS models.py x admin.py
myapp > models.py > Student > fees
1 from django.db import models
2
3 # Create your models here.
4 class Student(models.Model):
5     first_name = models.CharField(max_length=30)
6     last_name = models.CharField(max_length=30)
7     gender = models.BooleanField(default=False)
8     fees = models.IntegerField()
9
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Quit the server with CTRL-BREAK.
D:\lecture2k21\djangoproject\myapp\admin.py changed, reloading.
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
June 07, 2021 - 16:23:10
Django version 3.2.3, using settings 'djangoproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
In 8. Col 9 (4 selected) - Spaces: 4
```

Admin file



The screenshot shows the Visual Studio Code interface with the 'admin.py' file open in the editor. The code registers the 'Student' model with the Django admin site.

```
admin.py - djangoproject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER OPEN EDITORS models.py x admin.py
myapp > admin.py
1 from django.contrib import admin
2
3 # Register your models here.
4 from .models import Student
5
6 admin.site.register(Student)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Quit the server with CTRL-BREAK.
D:\lecture2k21\djangoproject\myapp\admin.py changed, reloading.
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
June 07, 2021 - 16:23:10
Django version 3.2.3, using settings 'djangoproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

0001_initial

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure:
 - DJANGOPROJECT
 - djangoproject
 - myapp
 - models.py
 - admin.py
 - 0001_initial.py
 - __pycache__
 - migrations
 - __pycache__
 - __init__.py
 - 0001_initial.py
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
 - static
 - templates
 - db.sqlite3
- Editor:** The file `0001_initial.py` is open, showing the following code:

```
myapp > migrations > 0001_initial.py > Migration
      from django.db import migrations, models
      ...
      class Migration(migrations.Migration):
          initial = True
          dependencies = [
          ]
          operations = [
              migrations.CreateModel(
                  name='Student',
                  fields=[
                      migrations.CharField(max_length=100)
                  ],
                  options={
                      'verbose_name': 'Student'
                  }
              )
          ]
      
```
- Terminal:** The terminal window shows the command `python manage.py makemigrations` being run, followed by the output:

```
D:\lecture2k21\djangoproject>python manage.py makemigrations
Migrations for 'myapp':
  myapp\migrations\0001_initial.py
    - Create model Student
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure, identical to the first screenshot.
- Editor:** The file `0001_initial.py` is open, showing the generated migration code:

```
myapp > migrations > 0001_initial.py > Migration
      # Generated by Django 3.2.3 on 2021-06-07 10:55
      from django.db import migrations, models
      ...
      class Migration(migrations.Migration):
          initial = True
          dependencies = [
          ]
          operations = [
              migrations.CreateModel(
                  name='Student',
                  fields=[
                      migrations.CharField(max_length=100)
                  ],
                  options={
                      'verbose_name': 'Student'
                  }
              )
          ]
      
```
- Terminal:** The terminal window shows the command `python manage.py makemigrations` being run again, but the output is identical to the previous run:

```
D:\lecture2k21\djangoproject>python manage.py makemigrations
Migrations for 'myapp':
  myapp\migrations\0001_initial.py
    - Create model Student
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure:
 - OPEN EDITORS: models.py (myapp), admin.py, 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
- Code Editor:** The file 0001_initial.py contains Python code for a database migration. The code defines a model named 'Student' with fields: id (BigAutoField), first_name (CharField), last_name (CharField), gender (BooleanField), and fees (IntegerField).

```
operations = [
    migrations.CreateModel(
        name='Student',
        fields=[
            ('id', models.BigAutoField(auto_created=True)),
            ('first_name', models.CharField(max_length=100)),
            ('last_name', models.CharField(max_length=100)),
            ('gender', models.BooleanField(default=False)),
            ('fees', models.IntegerField()),
        ],
    ),
]
```
- Bottom Navigation:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL.

Output:-

The screenshot shows the Django Admin interface at the URL `127.0.0.1:8000/admin/myapp/students/`. The page has the following structure:

- Header:** Django administration, Site administration.
- AUTHENTICATION AND AUTHORIZATION:** Groups, Users.
- MYAPP:** Students.
- Students List:** A table with columns: first_name, last_name, gender, fees, actions (Add, Change).

Django administration

WELCOME,

Home · Myapp · Students · Add student

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

MYAPP

Students [+ Add](#)

Add student

First name:

Last name:

Gender

Fees:

[Save and add another](#)

The screenshot shows the Django Admin interface for a 'Students' model. The sidebar on the left has a 'MYAPP' section with 'Students' highlighted in yellow. The main area is titled 'Add student' and contains fields for 'First name' (with 'Test' entered), 'Last name' (with 'Test' entered), a 'Gender' checkbox (unchecked), and 'Fees' (with '1' entered). A 'Save and add another' button is at the bottom right.