

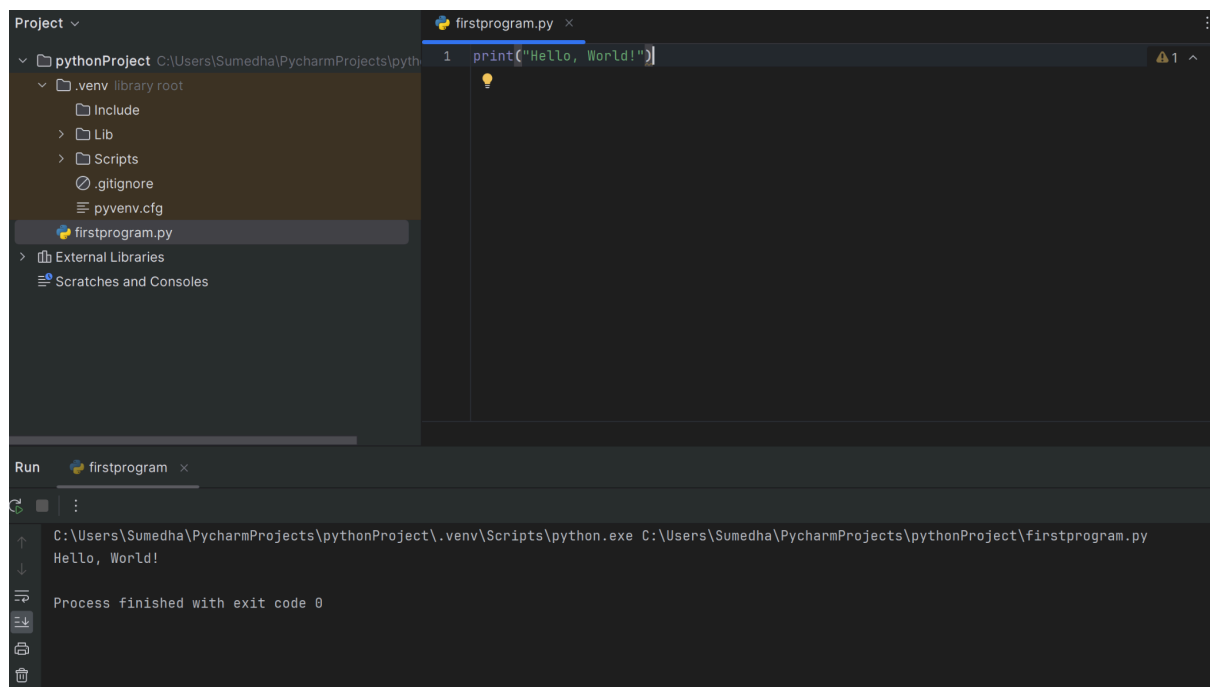
NAME: SHARATH CHANDRA

BATCH: DATA ENGINEERING

TOPIC: BASICS OF PYTHON

DATE : 25-01-24

1)Writing and Running First Program:

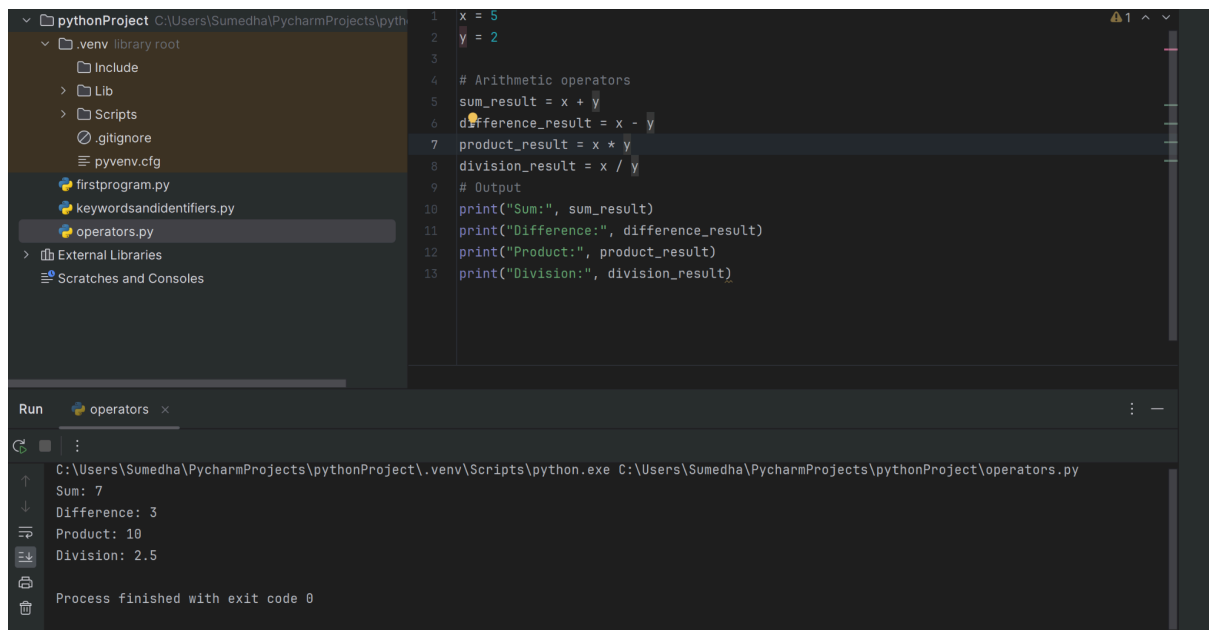


2)Keywords & Identifiers:

Python keywords

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

3) Variables & Operators:



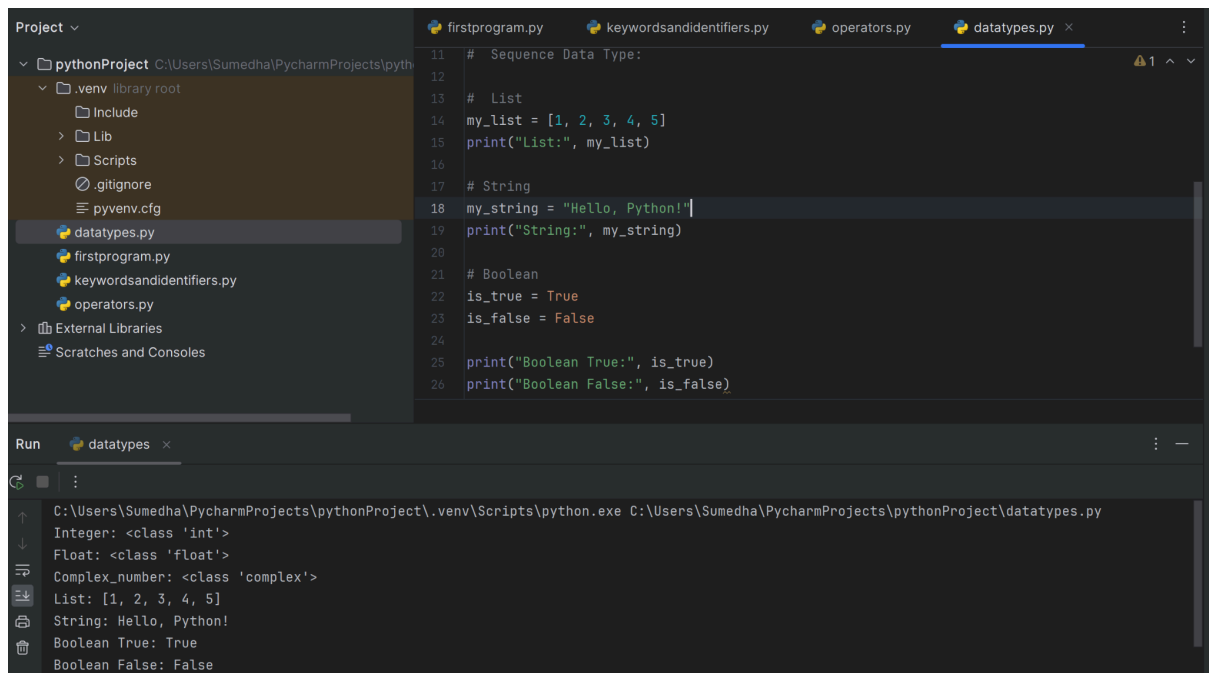
The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows the project structure, including a '.venv' directory and several Python files. The main editor displays a file named 'operators.py' with the following code:

```
1 x = 5
2 y = 2
3
4 # Arithmetic operators
5 sum_result = x + y
6 difference_result = x - y
7 product_result = x * y
8 division_result = x / y
9 # Output
10 print("Sum:", sum_result)
11 print("Difference:", difference_result)
12 print("Product:", product_result)
13 print("Division:", division_result)
```

The Run window at the bottom shows the execution output for 'operators.py':

```
Sum: 7
Difference: 3
Product: 10
Division: 2.5
Process finished with exit code 0
```

4) Data Types:



The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows the project structure, including a '.venv' directory and several Python files. The main editor displays a file named 'datatypes.py' with the following code:

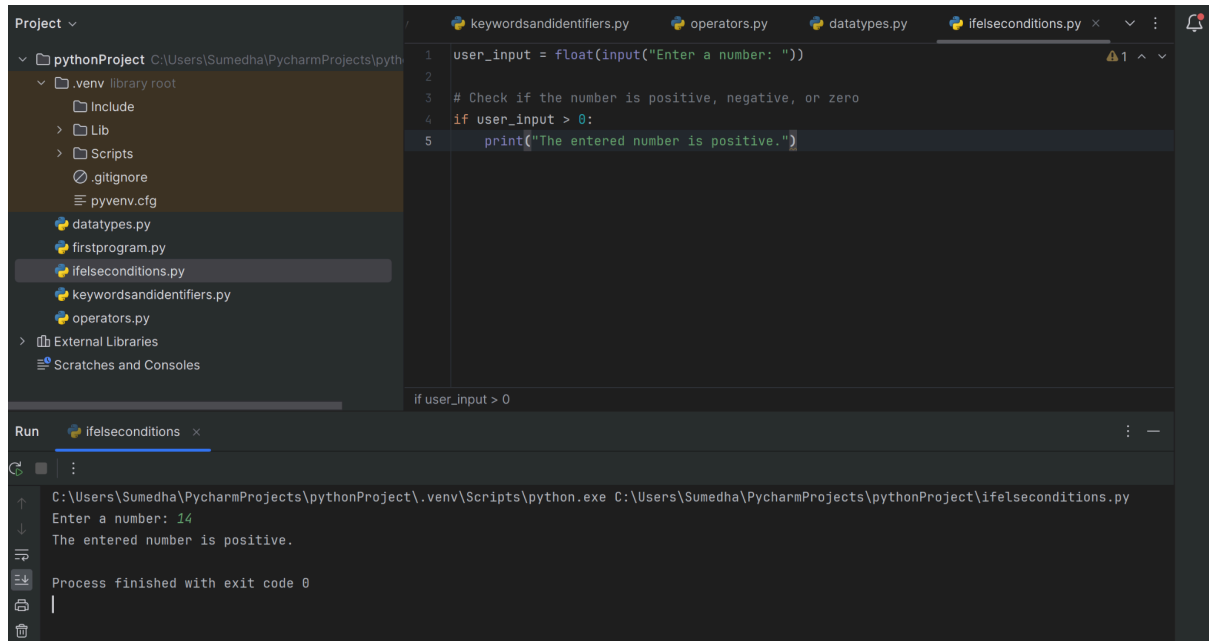
```
11 # Sequence Data Type:
12
13 # List
14 my_list = [1, 2, 3, 4, 5]
15 print("List:", my_list)
16
17 # String
18 my_string = "Hello, Python!"
19 print("String:", my_string)
20
21 # Boolean
22 is_true = True
23 is_false = False
24
25 print("Boolean True:", is_true)
26 print("Boolean False:", is_false)
```

The Run window at the bottom shows the execution output for 'datatypes.py':

```
Integer: <class 'int'>
Float: <class 'float'>
Complex_number: <class 'complex'>
List: [1, 2, 3, 4, 5]
String: Hello, Python!
Boolean True: True
Boolean False: False
```

5) Conditional statements:

i) IF Condition:

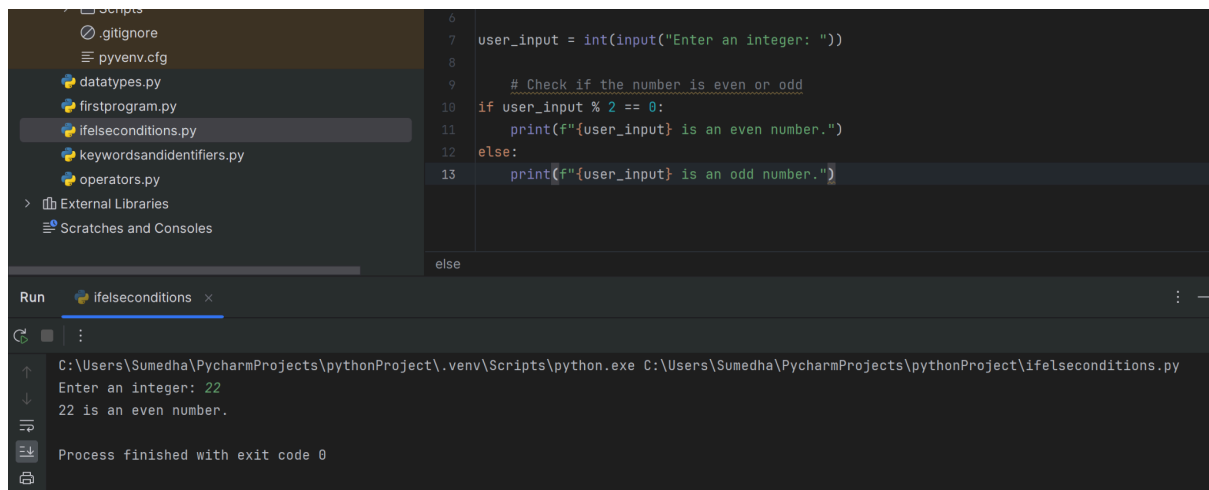


The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left lists several Python files, including 'ifelseconditions.py' which is currently selected. The main editor window displays the code for 'ifelseconditions.py' with the following content:

```
1 user_input = float(input("Enter a number: "))
2
3 # Check if the number is positive, negative, or zero
4 if user_input > 0:
5     print("The entered number is positive.")
```

Below the editor, the 'Run' console shows the execution of the program. It prompts 'Enter a number: 14' and outputs 'The entered number is positive.'. The console also indicates 'Process finished with exit code 0'.

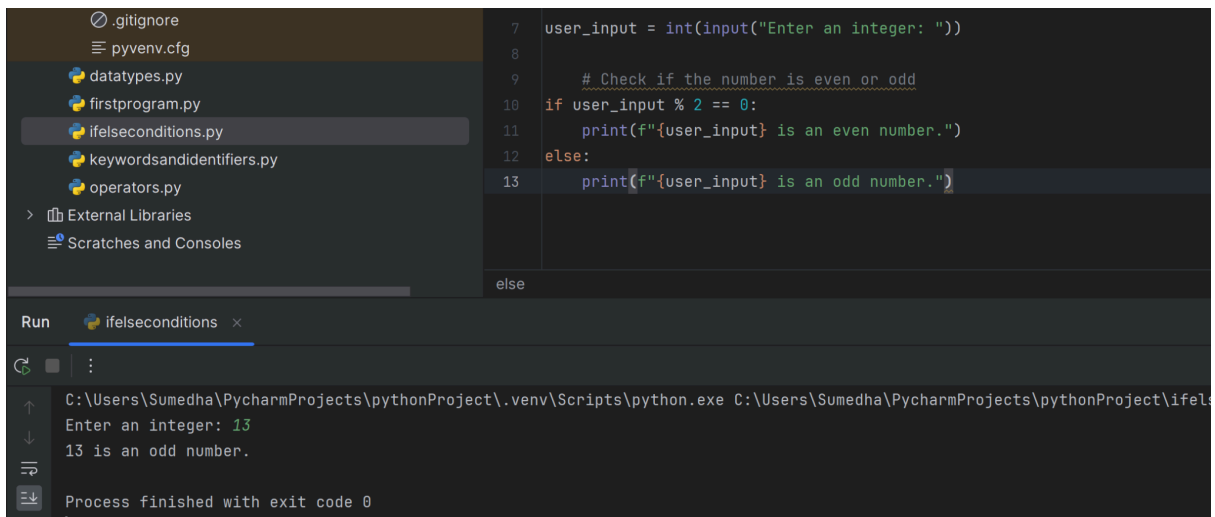
ii) IF - Else Condition:



The screenshot shows the PyCharm IDE with the same project. The file explorer on the left shows 'ifelseconditions.py' selected. The main editor window displays the code for 'ifelseconditions.py' with the following content:

```
6
7 user_input = int(input("Enter an integer: "))
8
9 # Check if the number is even or odd
10 if user_input % 2 == 0:
11     print(f"{user_input} is an even number.")
12 else:
13     print(f"{user_input} is an odd number.")
```

Below the editor, the 'Run' console shows the execution of the program. It prompts 'Enter an integer: 22' and outputs '22 is an even number.'. The console also indicates 'Process finished with exit code 0'.



```
7 user_input = int(input("Enter an integer: "))
8
9 # Check if the number is even or odd
10 if user_input % 2 == 0:
11     print(f"{user_input} is an even number.")
12 else:
13     print(f"{user_input} is an odd number.")
```

Run ifelseconditions x

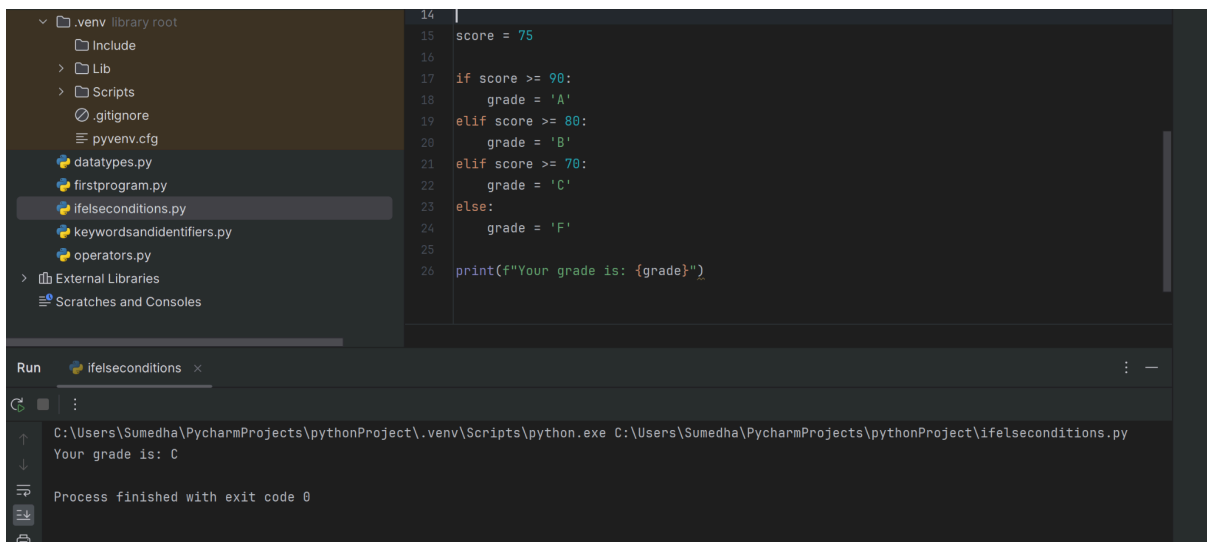
C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\ifelseconditions.py

Enter an integer: 13

13 is an odd number.

Process finished with exit code 0

iii) IF - Elif - Else Condition:



```
14
15 score = 75
16
17 if score >= 90:
18     grade = 'A'
19 elif score >= 80:
20     grade = 'B'
21 elif score >= 70:
22     grade = 'C'
23 else:
24     grade = 'F'
25
26 print(f"Your grade is: {grade}")
```

Run ifelseconditions x

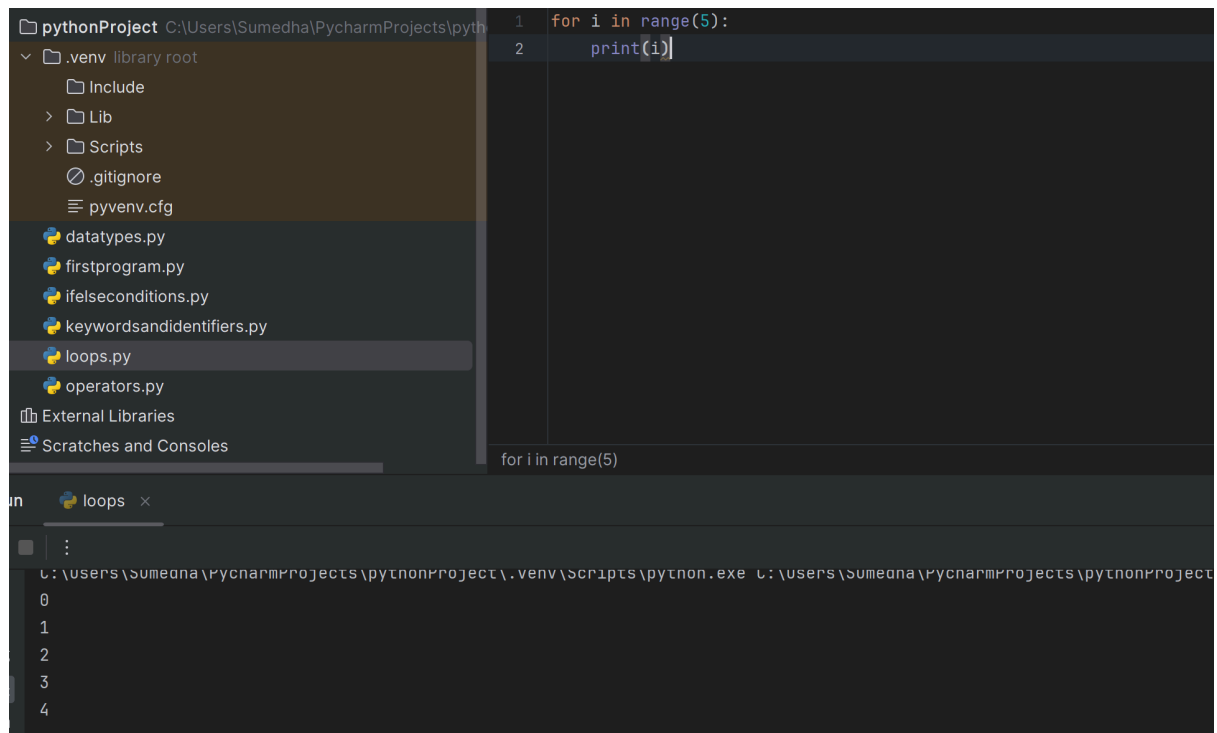
C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\ifelseconditions.py

Your grade is: C

Process finished with exit code 0

6) Loops:

i) for loop:

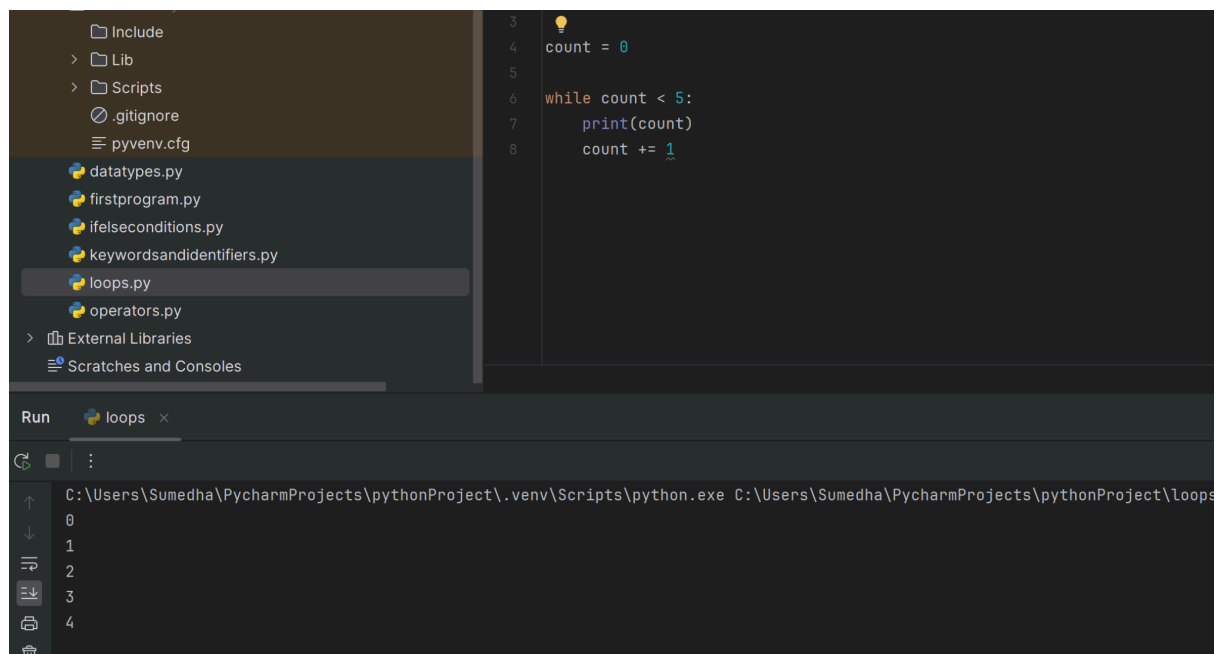


The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left lists several Python files, with 'loops.py' selected. The main editor displays the following code in 'loops.py':

```
1 for i in range(5):  
2     print(i)
```

The Run console at the bottom shows the output of the program, displaying the numbers 0 through 4, each on a new line, indicating that the loop executed successfully.

ii) While loop:

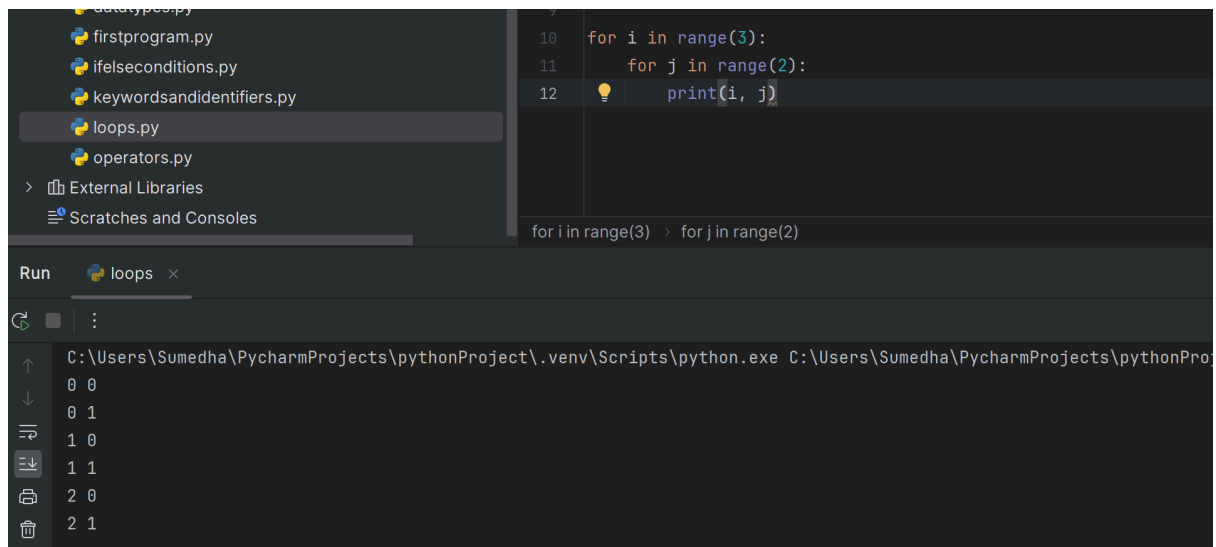


The screenshot shows the PyCharm IDE with the same project. The file explorer on the left lists several Python files, with 'loops.py' selected. The main editor displays the following code in 'loops.py':

```
3  
4 count = 0  
5  
6 while count < 5:  
7     print(count)  
8     count += 1
```

The Run console at the bottom shows the output of the program, displaying the numbers 0 through 4, each on a new line, indicating that the loop executed successfully.

iii) Nested For loop:



```
10 for i in range(3):
11     for j in range(2):
12         print(i, j)
```

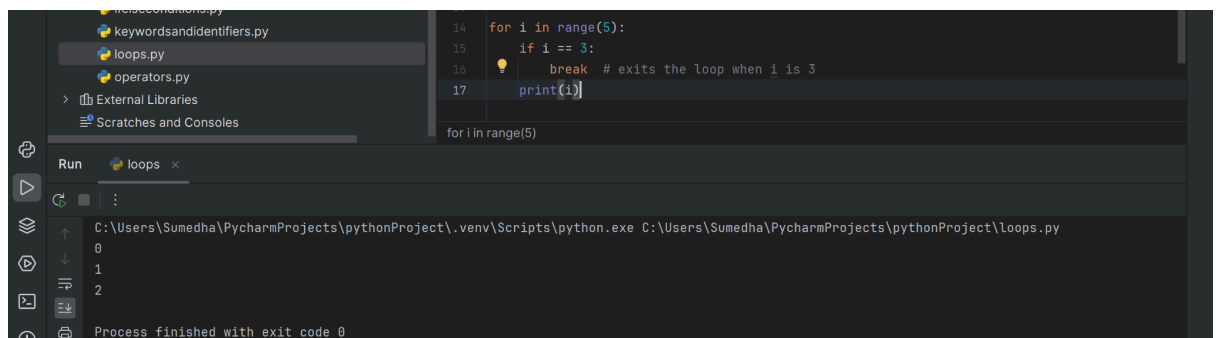
for i in range(3) > for j in range(2)

Run loops x

C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\loops.py

0 0
0 1
1 0
1 1
2 0
2 1

iv) Break:



```
14 for i in range(5):
15     if i == 3:
16         break # exits the loop when i is 3
17     print(i)
```

for i in range(5)

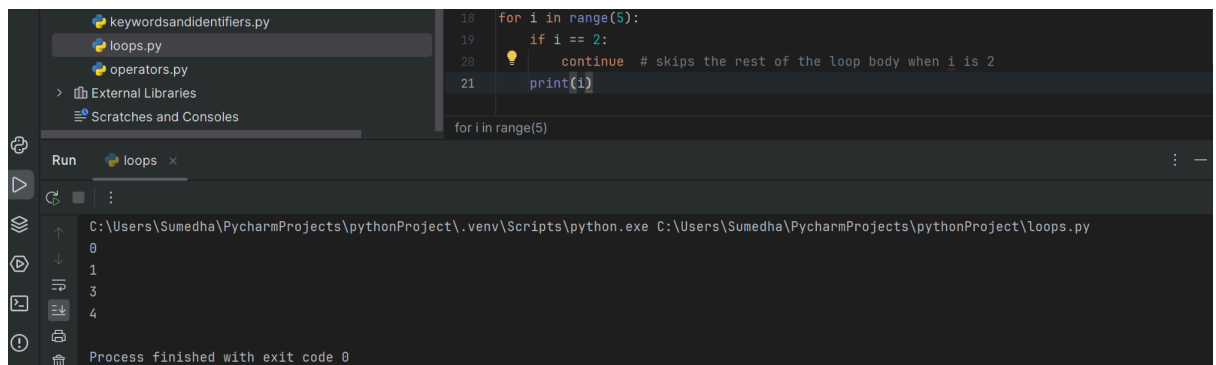
Run loops x

C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\loops.py

0
1
2

Process finished with exit code 0

v) Continue:



```
18 for i in range(5):
19     if i == 2:
20         continue # skips the rest of the loop body when i is 2
21     print(i)
```

for i in range(5)

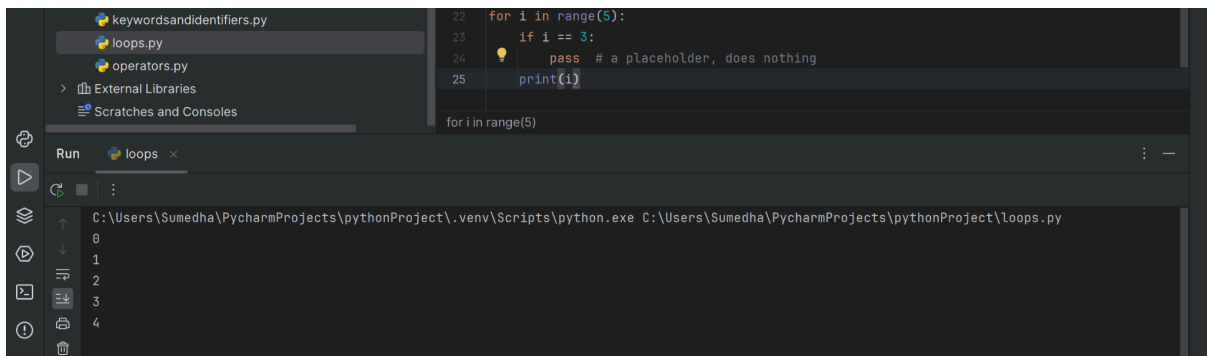
Run loops x

C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\loops.py

0
1
3
4

Process finished with exit code 0

vi)pass:

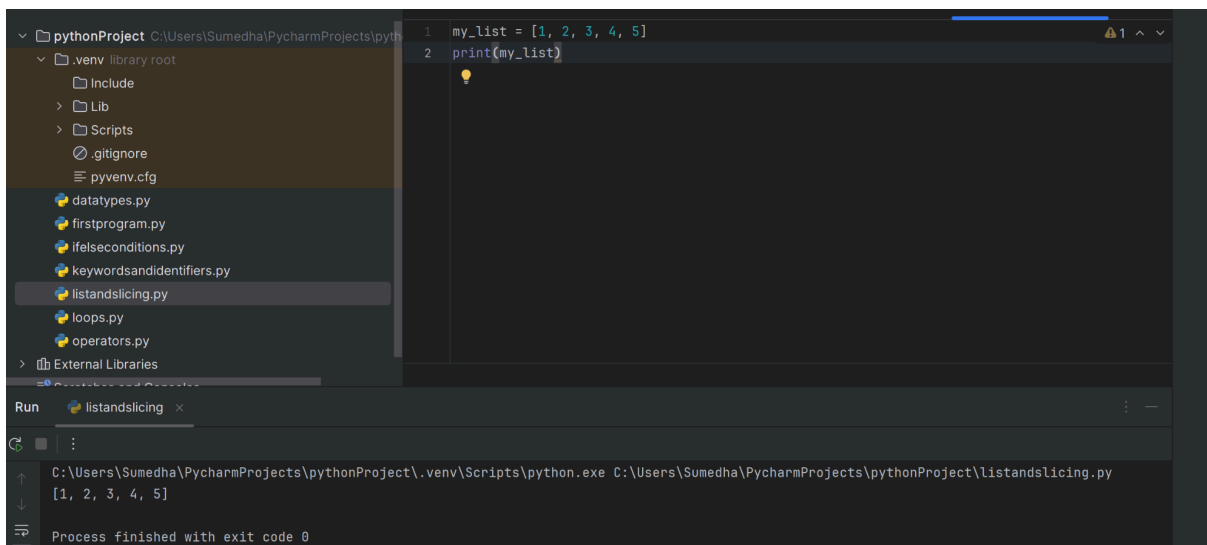


The image shows the PyCharm IDE interface. The left sidebar displays a project tree with files: keywordsandidentifiers.py, loops.py, operators.py, External Libraries, and Scratches and Consoles. The main editor window shows a Python script with the following code:

```
22 for i in range(5):
23     if i == 3:
24         pass # a placeholder, does nothing
25     print(i)
```

Below the editor, the Run tab is active, showing the command: `C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\loops.py`. The output console displays the numbers 0, 1, 2, 3, and 4, each on a new line.

7)Lists:



The image shows the PyCharm IDE interface. The left sidebar displays a project tree with files: pythonProject, .venv, library root, Include, Lib, Scripts, .gitignore, pyvenv.cfg, datatypes.py, firstprogram.py, ifelseconditions.py, keywordsandidentifiers.py, listandslicing.py, loops.py, operators.py, External Libraries, and Scratches and Consoles. The main editor window shows a Python script with the following code:

```
1 my_list = [1, 2, 3, 4, 5]
2 print(my_list)
```

Below the editor, the Run tab is active, showing the command: `C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\listandslicing.py`. The output console displays the list `[1, 2, 3, 4, 5]`. At the bottom, it says "Process finished with exit code 0".

i)List Methods and Slicing:

```
fruits = ['apple', 'banana', 'orange']
fruits.append('grape')
print(fruits[1:3])

#append
numbers = [1, 2, 3]
numbers.append(4)
print(numbers) # Output: [1, 2, 3, 4]

#extend
list1 = [1, 2, 3]
list2 = [4, 5, 6]
list1.extend(list2)
print(list1) # Output: [1, 2, 3, 4, 5, 6]

#insert
fruits = ['apple', 'banana', 'cherry']
fruits.insert(_index: 1, _object: 'orange')
print(fruits) # Output: ['apple', 'orange', 'banana', 'cherry']

#remove
numbers = [1, 2, 3, 4, 3]
numbers.remove(3)
print(numbers) # Output: [1, 2, 4, 3]
```

```
#pop
numbers = [1, 2, 3, 4]
popped_element = numbers.pop(2)
print(popped_element) # Output: 3
print(numbers) # Output: [1, 2, 4]

#index
numbers = [10, 20, 30, 20, 40]
index = numbers.index(20)
print(index) # Output: 1
```

Output:

```
C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\listandslicing.py
['banana', 'orange']
[1, 2, 3, 4]
[1, 2, 3, 4, 5, 6]
['apple', 'orange', 'banana', 'cherry']
[1, 2, 4, 3]
3
[1, 2, 4]
1
Process finished with exit code 0
```


8)Dictionaries & Dictionary Methods:

```
pythonProject C:\Users\Sumedha\
├── .venv library root
│   ├── Include
│   ├── Lib
│   ├── Scripts
│   ├── .gitignore
│   └── pyvenv.cfg
├── datatypes.py
├── DictionariesDictionaryMethods.p
├── firstprogram.py
├── ifelseconditions.py
├── keywordsandidentifiers.py
├── listandslicing.py
├── loops.py
├── operators.py
├── External Libraries
└── Scratches and Consoles

1 student = {'name': 'John', 'age': 20, 'grade': 'B'}
2 print(student['name'])
3 # get(key, default)
4 student = {'name': 'John', 'age': 20, 'grade': 'B'}
5 age = student.get('age', 'N/A')
6 print(age) # Output: 20
7
8 # Using get with a key that doesn't exist
9 address = student.get('address', 'N/A')
10 print(address) # Output: N/A
11
12 # keys()
13
14 student = {'name': 'John', 'age': 20, 'grade': 'B'}
15 all_keys = student.keys()
16 print(all_keys)
17
18 # values()
19
20 student = {'name': 'John', 'age': 20, 'grade': 'B'}
21 all_values = student.values()
22 print(all_values)
23
24 # items()
25 student = {'name': 'John', 'age': 20, 'grade': 'B'}
26 all_items = student.items()
27 print(all_items)
```

Output:

```
Run listandslicing x
C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\listandslicing.py
['banana', 'orange']
[1, 2, 3, 4]
[1, 2, 3, 4, 5, 6]
['apple', 'orange', 'banana', 'cherry']
[1, 2, 4, 3]
3
[1, 2, 4]
1
Process finished with exit code 0
```

9)Set & Set Methods:

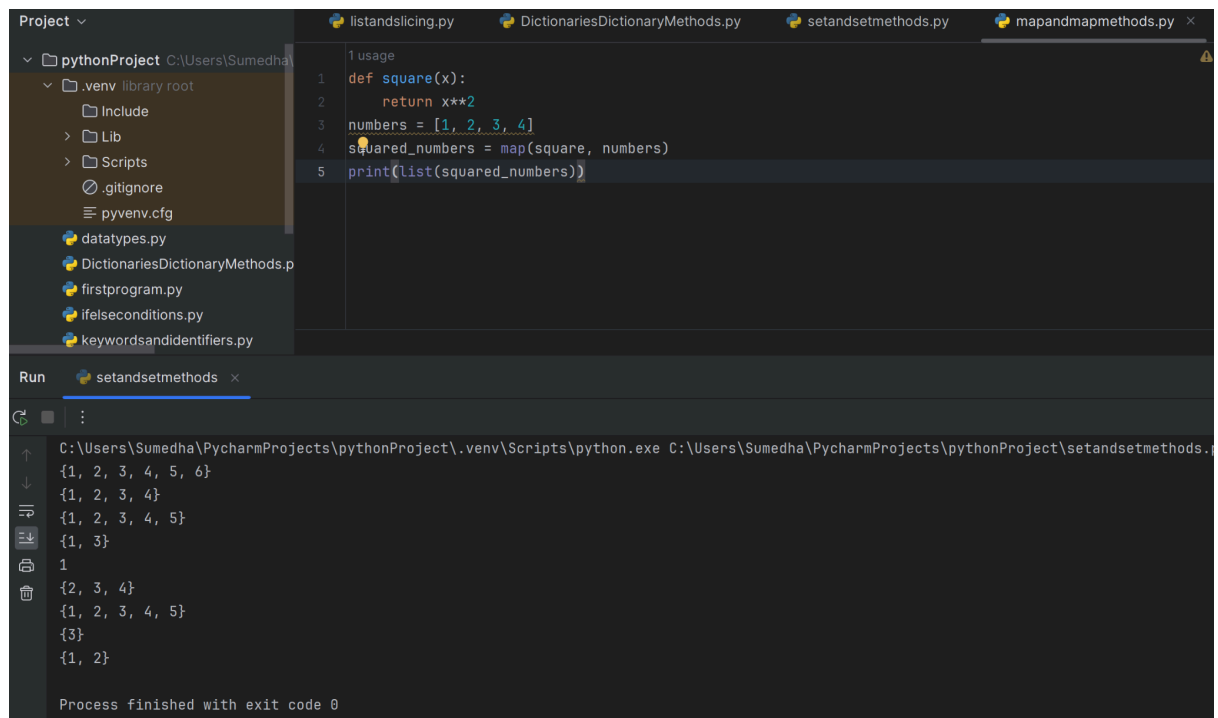
```
pythonProject C:\Users\Sumedha\
├── .venv library root
│   ├── Include
│   ├── Lib
│   ├── Scripts
│   ├── .gitignore
│   └── pyvenv.cfg
├── datatypes.py
├── DictionariesDictionaryMethods.p
├── firstprogram.py
├── ifelseconditions.py
├── keywordsandidentifiers.py
├── listandslicing.py
├── loops.py
├── operators.py
└── setandsetmethods.py
External Libraries
Scratches and Consoles

1 my_set = {1, 2, 3, 4, 5}
2 my_set.add(6)
3 print(my_set)
4 # add(element)
5 my_set = {1, 2, 3}
6 my_set.add(4)
7 print(my_set)
8 # update(iterable)
9 set1 = {1, 2, 3}
10 set2 = {3, 4, 5}
11 set1.update(set2)
12 print(set1)
13 # remove(element)
14 my_set = {1, 2, 3}
15 my_set.remove(2)
16 print(my_set)
17 # pop()
18 my_set = {1, 2, 3, 4}
19 popped_element = my_set.pop()
20 print(popped_element)
21 print(my_set)
22 # union(iterable)
23 set1 = {1, 2, 3}
24 set2 = {3, 4, 5}
25 union_set = set1.union(set2)
26 # Alternatively: union_set = set1 | set2
27 print(union_set)
28
29 print(union_set)
30 # intersection(iterable)
31 set1 = {1, 2, 3}
32 set2 = {3, 4, 5}
33 intersection_set = set1.intersection(set2)
34 # Alternatively: intersection_set = set1 & set2
35 print(intersection_set)
36 # difference(iterable)
37 set1 = {1, 2, 3}
38 set2 = {3, 4, 5}
39 difference_set = set1.difference(set2)
40 # Alternatively: difference_set = set1 - set2
41 print(difference_set)
```

Output:

```
C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\setandsetmethods.py
{1, 2, 3, 4, 5, 6}
{1, 2, 3, 4}
{1, 2, 3, 4, 5}
{1, 3}
1
{2, 3, 4}
{1, 2, 3, 4, 5}
{3}
{1, 2}
Process finished with exit code 0
```

10) Introduction to Map & Map Methods:



The screenshot shows the PyCharm IDE interface. The top toolbar contains icons for 'listandslicing.py', 'DictionariesDictionaryMethods.py', 'setandsetmethods.py', and 'mapandmapmethods.py'. The left sidebar displays the 'Project' view for 'pythonProject' at 'C:\Users\Sumedha\...', showing a file tree with '.venv' (library root, Include, Lib, Scripts), '.gitignore', 'pyvenv.cfg', and several Python files including 'datatypes.py', 'DictionariesDictionaryMethods.p', 'firstprogram.py', 'ifelseconditions.py', and 'keywordsandidentifiers.py'. The main editor window shows the code in 'setandsetmethods.py':

```
1 usage
2 def square(x):
3     return x**2
4 numbers = [1, 2, 3, 4]
5 squared_numbers = map(square, numbers)
6 print(list(squared_numbers))
```

The bottom 'Run' panel shows the execution of 'setandsetmethods.py'. The command line is: `C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\setandsetmethods.py`. The output is:

```
{1, 2, 3, 4, 5, 6}
{1, 2, 3, 4}
{1, 2, 3, 4, 5}
{1, 3}
1
{2, 3, 4}
{1, 2, 3, 4, 5}
{3}
{1, 2}
```

Process finished with exit code 0