

Name : Parth Nandedkar

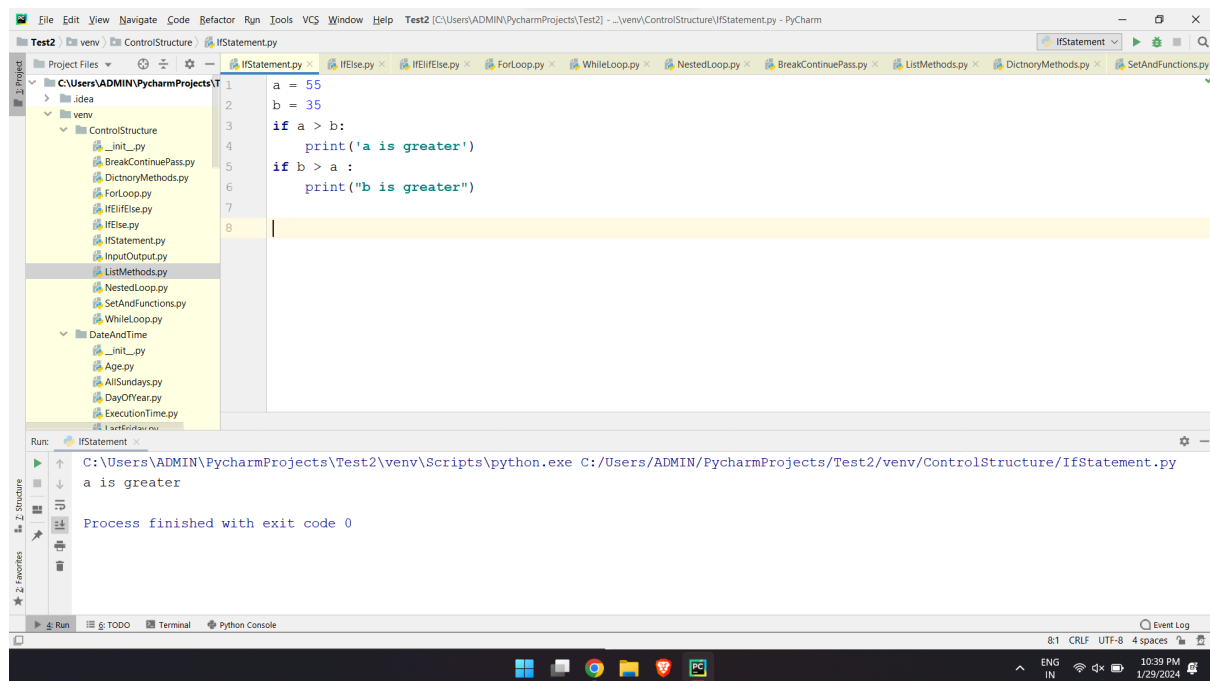
Date : 29 Jan 2024

Topics : Python

Batch : Data Engineering Batch-1

Control Structure :

If statement :



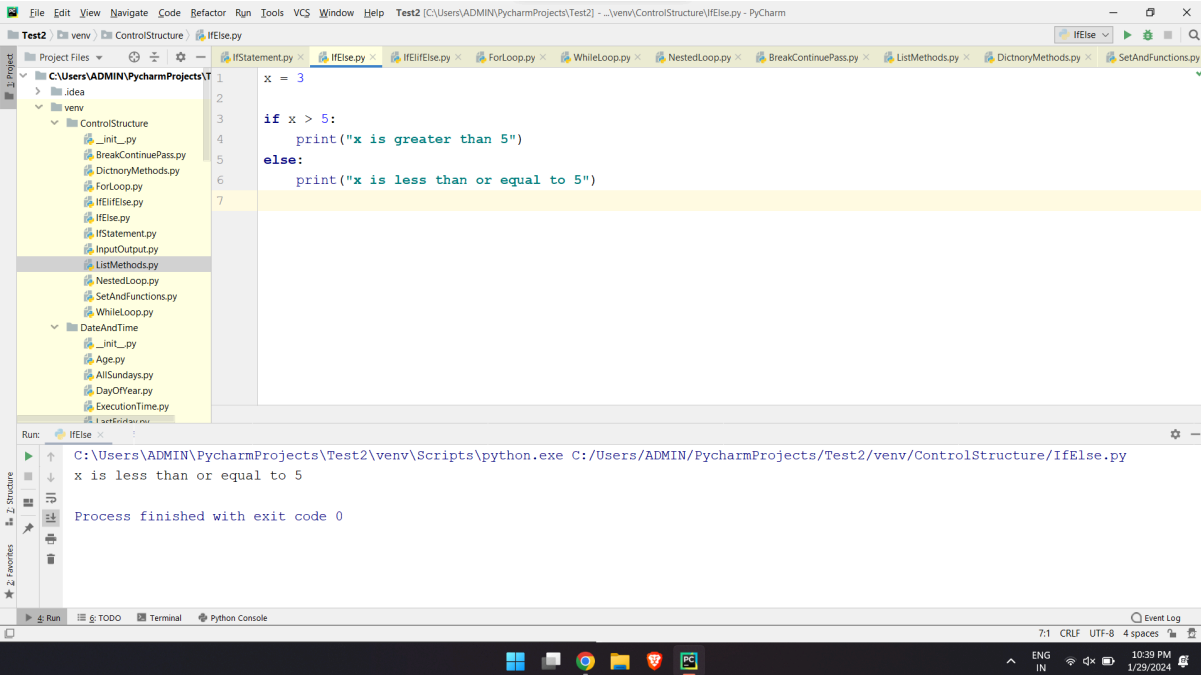
The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `IfStatement.py` with the following code:

```
1 a = 55
2 b = 35
3 if a > b:
4     print('a is greater')
5 if b > a :
6     print("b is greater")
7
8
```

The left sidebar shows the project structure, including a `ControlStructure` folder containing various Python files. The bottom panel shows the Run output, indicating that the program executed successfully and finished with exit code 0.

```
Run: IfStatement
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/ControlStructure/IfStatement.py
a is greater
Process finished with exit code 0
```

If else :



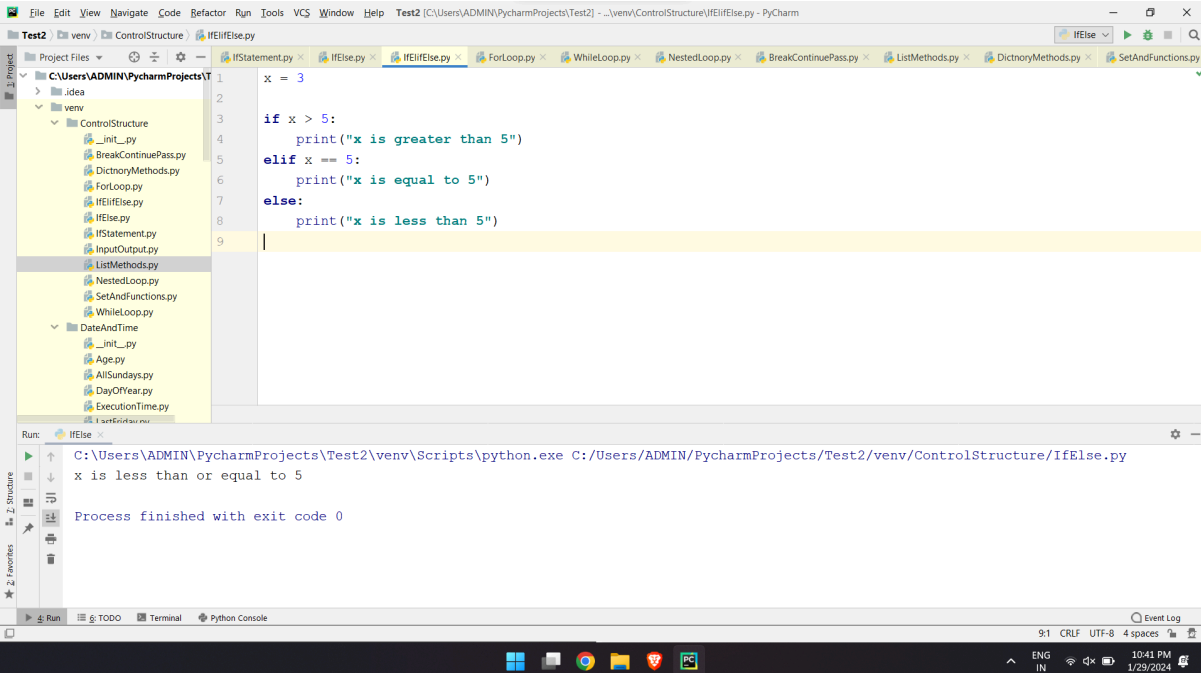
```
1 x = 3
2
3
4 if x > 5:
5     print("x is greater than 5")
6 else:
7     print("x is less than or equal to 5")
```

Run: IfElse

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/ControlStructure/IfElse.py
x is less than or equal to 5

Process finished with exit code 0
```

If elif else :



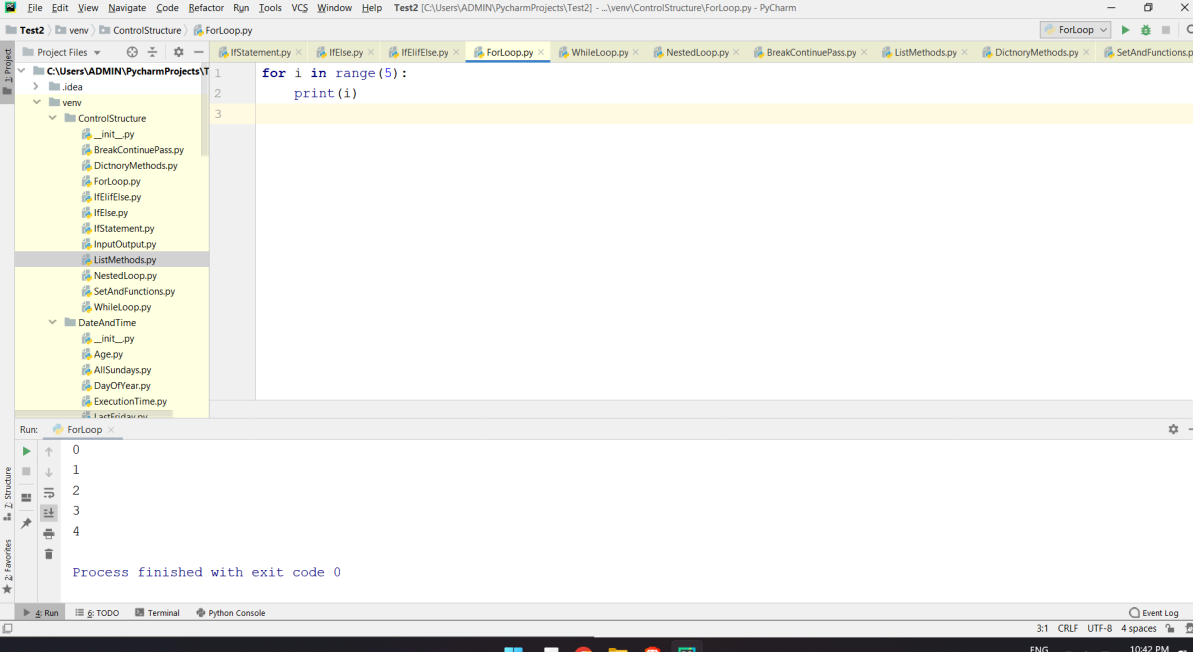
```
1 x = 3
2
3
4 if x > 5:
5     print("x is greater than 5")
6 elif x == 5:
7     print("x is equal to 5")
8 else:
9     print("x is less than 5")
```

Run: IfElse

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/ControlStructure/IfElse.py
x is less than or equal to 5

Process finished with exit code 0
```

For loop :



The screenshot shows the PyCharm IDE with a project named 'Test2'. The file explorer on the left shows a directory structure with 'ControlStructure' and 'DateTime' folders. The main editor displays the code for 'ForLoop.py':

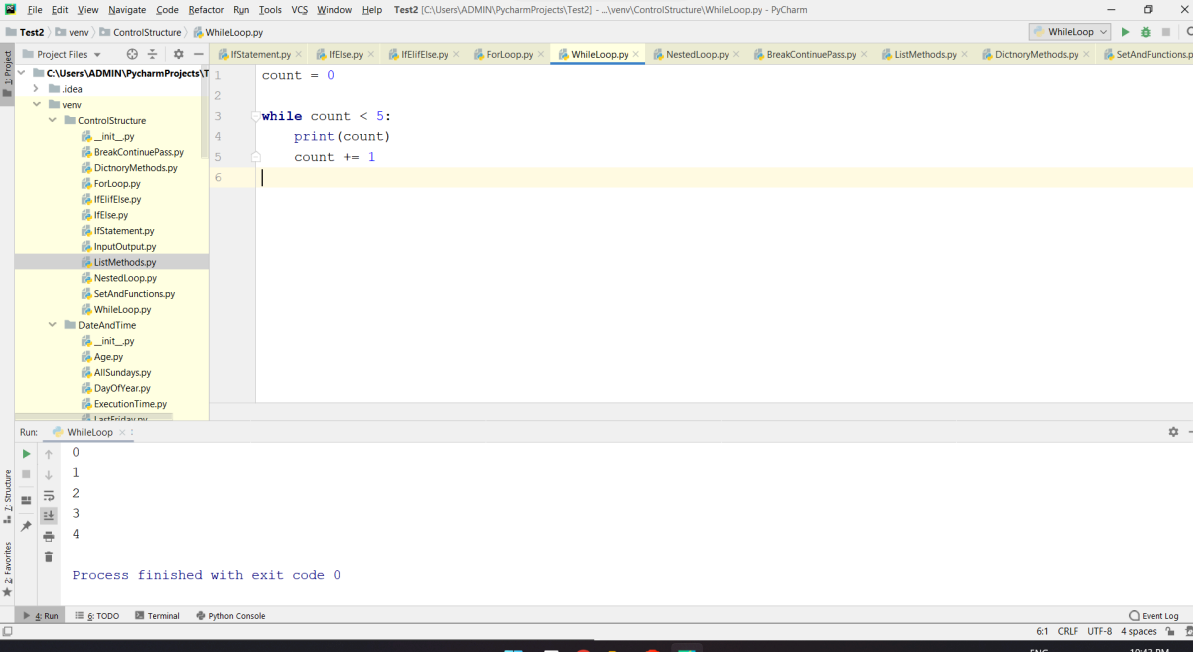
```
1 for i in range(5):  
2     print(i)  
3
```

The Run window at the bottom shows the execution output:

```
0  
1  
2  
3  
4  
  
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8 and the line length is 31 characters.

While loop :



The screenshot shows the PyCharm IDE with a project named 'Test2'. The file explorer on the left shows a directory structure with 'ControlStructure' and 'DateTime' folders. The main editor displays the code for 'WhileLoop.py':

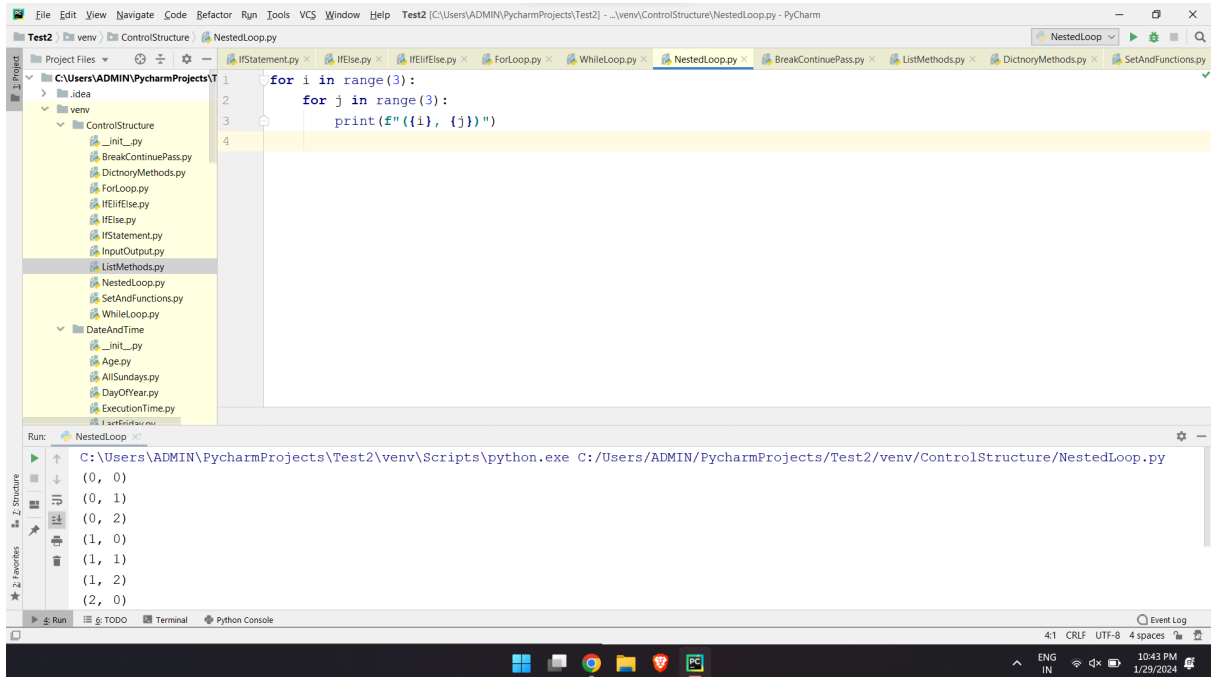
```
1 count = 0  
2  
3 while count < 5:  
4     print(count)  
5     count += 1  
6
```

The Run window at the bottom shows the execution output:

```
0  
1  
2  
3  
4  
  
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8 and the line length is 61 characters.

Nested loop :



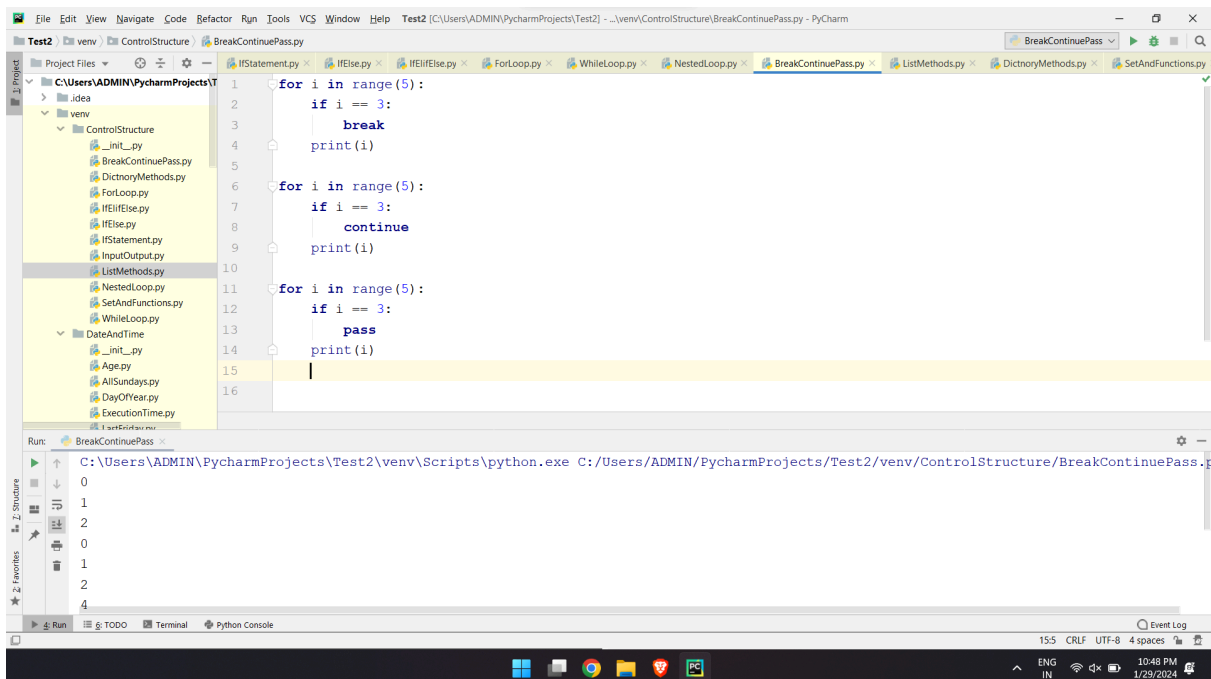
The screenshot shows the PyCharm IDE with a project named 'Test2'. The file explorer on the left shows a directory structure with a 'ControlStructure' folder containing several Python files, including 'NestedLoop.py'. The main editor window displays the code for 'NestedLoop.py':

```
1 for i in range(3):
2     for j in range(3):
3         print(f"({i}, {j})")
4
```

The Run console at the bottom shows the output of the program:

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/ControlStructure/NestedLoop.py
(0, 0)
(0, 1)
(0, 2)
(1, 0)
(1, 1)
(1, 2)
(2, 0)
```

Break, Continue, Pass :



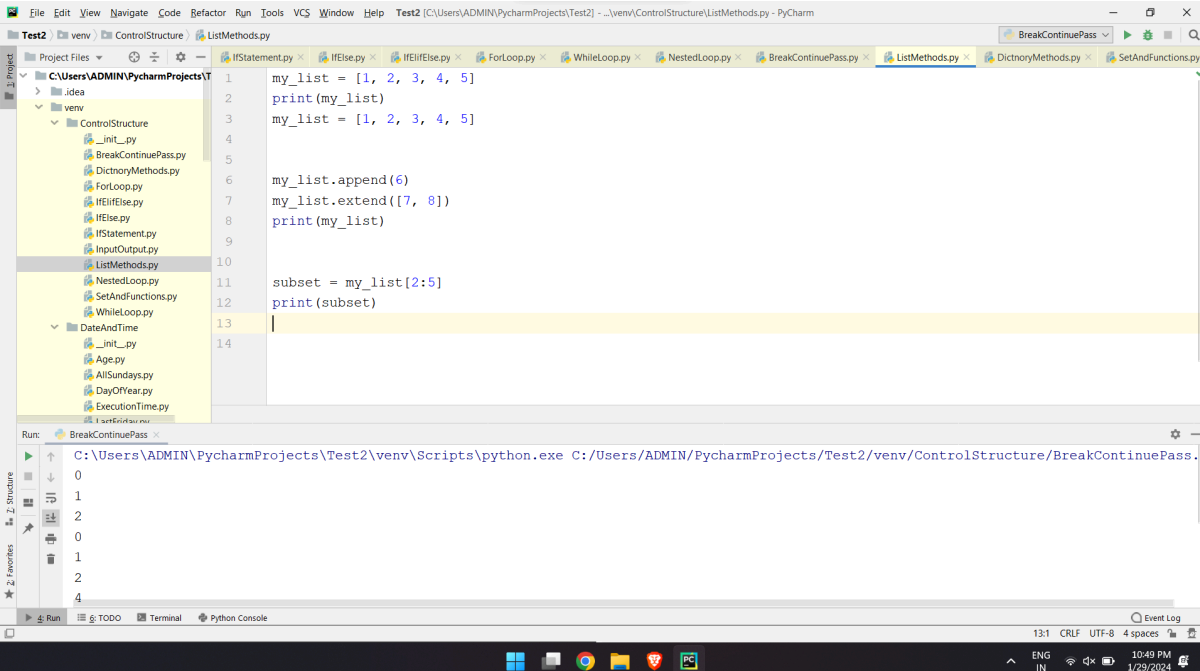
The screenshot shows the PyCharm IDE with a project named 'Test2'. The file explorer on the left shows a directory structure with a 'ControlStructure' folder containing several Python files, including 'BreakContinuePass.py'. The main editor window displays the code for 'BreakContinuePass.py':

```
1 for i in range(5):
2     if i == 3:
3         break
4     print(i)
5
6 for i in range(5):
7     if i == 3:
8         continue
9     print(i)
10
11 for i in range(5):
12     if i == 3:
13         pass
14     print(i)
15
16
```

The Run console at the bottom shows the output of the program:

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/ControlStructure/BreakContinuePass.py
0
1
2
0
1
2
4
```

List Methods :



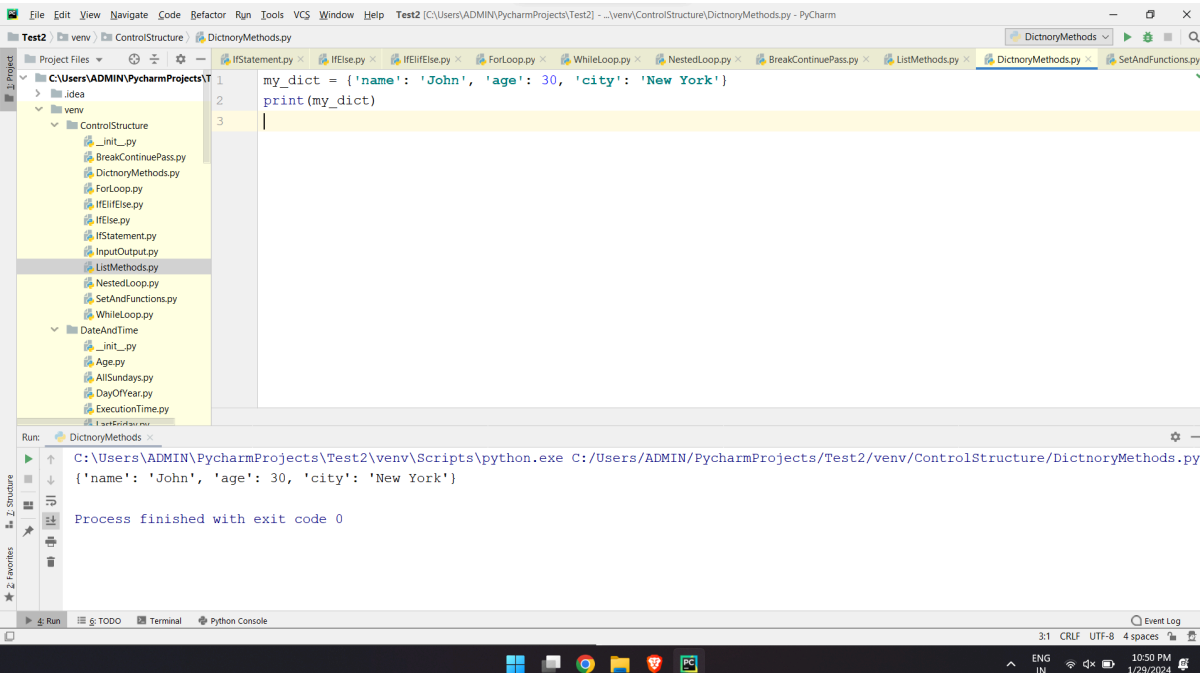
The screenshot shows the PyCharm IDE with the file `ListMethods.py` open. The code defines a list `my_list` and performs several operations on it. The terminal output shows the execution of the code, displaying the list and its subset.

```
1 my_list = [1, 2, 3, 4, 5]
2 print(my_list)
3 my_list = [1, 2, 3, 4, 5]
4
5
6 my_list.append(6)
7 my_list.extend([7, 8])
8 print(my_list)
9
10
11 subset = my_list[2:5]
12 print(subset)
13
14
```

Run: BreakContinuePass

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/ControlStructure/BreakContinuePass.py
0
1
2
0
1
2
4
```

Dictionary Methods :



The screenshot shows the PyCharm IDE with the file `DictnoryMethods.py` open. The code creates a dictionary `my_dict` and prints it. The terminal output shows the execution of the code, displaying the dictionary and the message "Process finished with exit code 0".

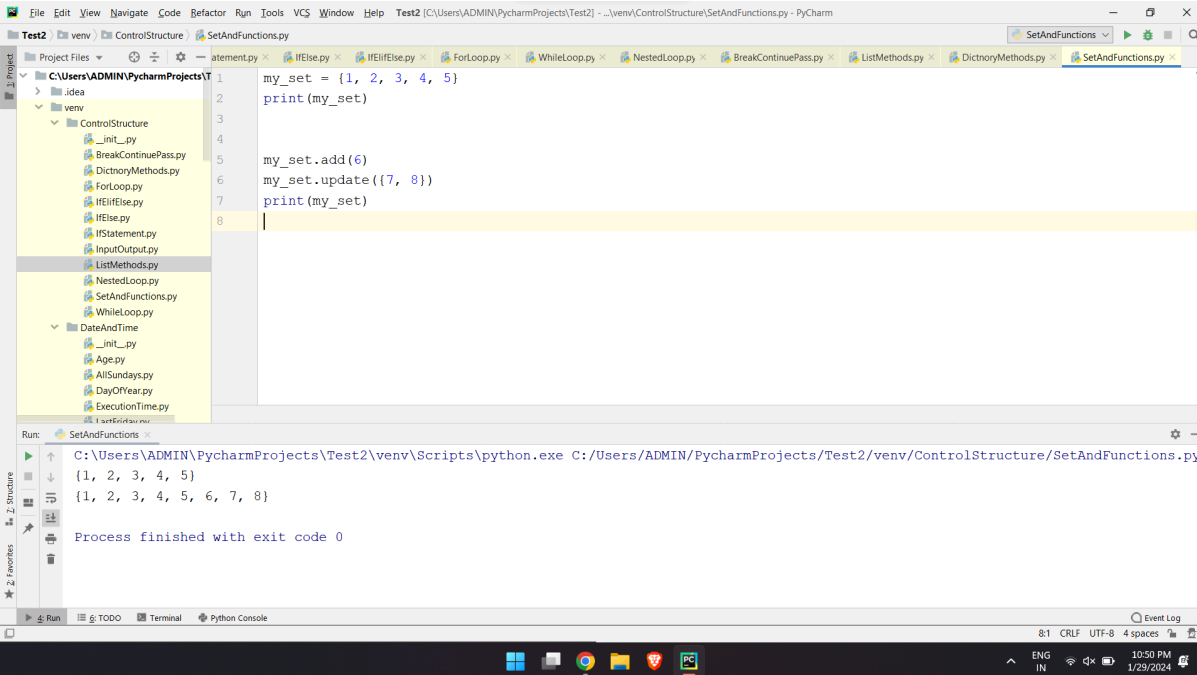
```
1 my_dict = {'name': 'John', 'age': 30, 'city': 'New York'}
2 print(my_dict)
3
```

Run: DictnoryMethods

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/ControlStructure/DictnoryMethods.py
{'name': 'John', 'age': 30, 'city': 'New York'}

Process finished with exit code 0
```

Set functions :



The screenshot displays the PyCharm IDE interface. The main editor window shows the file `SetAndFunctions.py` with the following Python code:

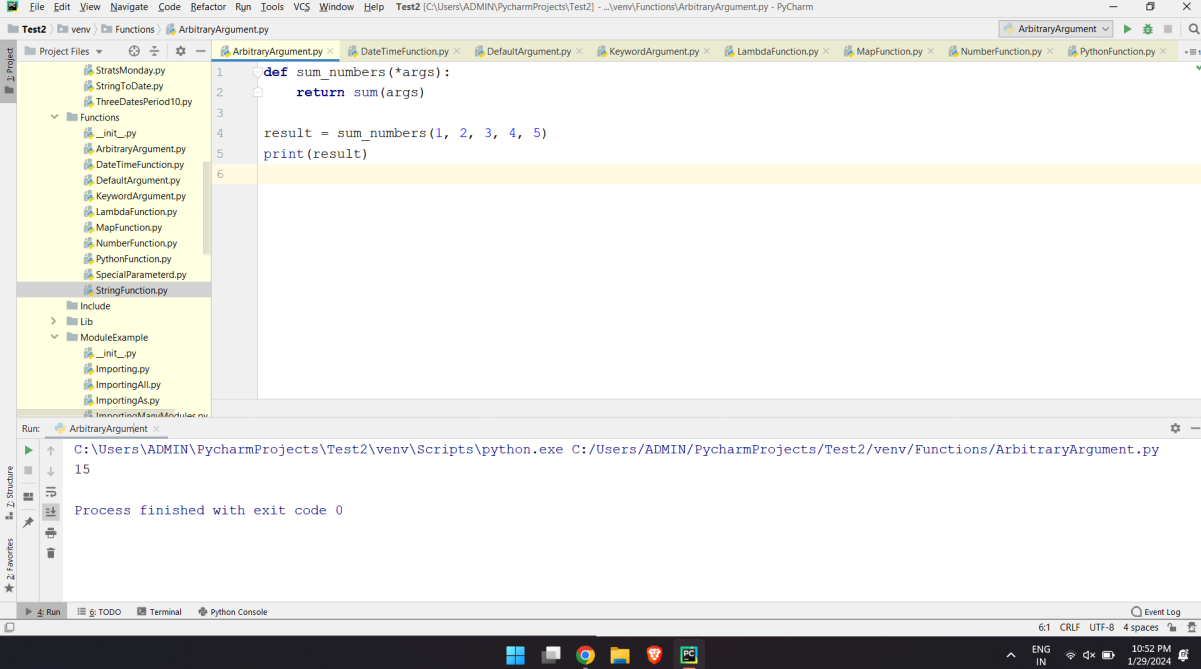
```
1 my_set = {1, 2, 3, 4, 5}
2 print(my_set)
3
4
5 my_set.add(6)
6 my_set.update({7, 8})
7 print(my_set)
8
```

The left sidebar shows the project structure for `Test2`, with the `ControlStructure` folder expanded, listing various Python files including `SetAndFunctions.py`.

The bottom panel shows the Run output for `SetAndFunctions.py`. The command executed is `C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:\Users\ADMIN\PycharmProjects\Test2\venv\ControlStructure/SetAndFunctions.py`. The output shows the initial set `{1, 2, 3, 4, 5}` and the updated set `{1, 2, 3, 4, 5, 6, 7, 8}` after the `add` and `update` operations. The process finished with exit code 0.

Functions :

Arbitrary constant :

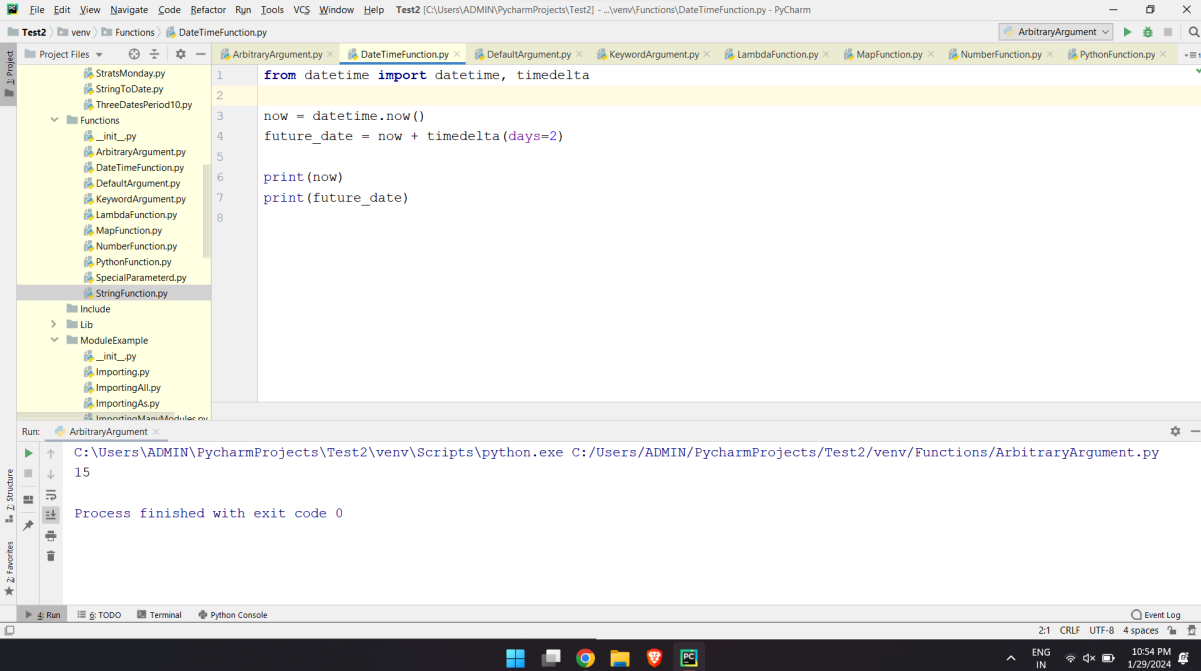


The screenshot shows the PyCharm IDE with the file `ArbitraryArgument.py` open. The code defines a function `sum_numbers(*args)` that returns the sum of its arguments. The function is called with `sum_numbers(1, 2, 3, 4, 5)`, and the result `15` is printed. The Run console shows the command `C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Functions/ArbitraryArgument.py` and the output `15`. The process finished with exit code 0.

```
1 def sum_numbers(*args):
2     return sum(args)
3
4 result = sum_numbers(1, 2, 3, 4, 5)
5 print(result)
6
```

Run: C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Functions/ArbitraryArgument.py
15
Process finished with exit code 0

Datetime Functions :

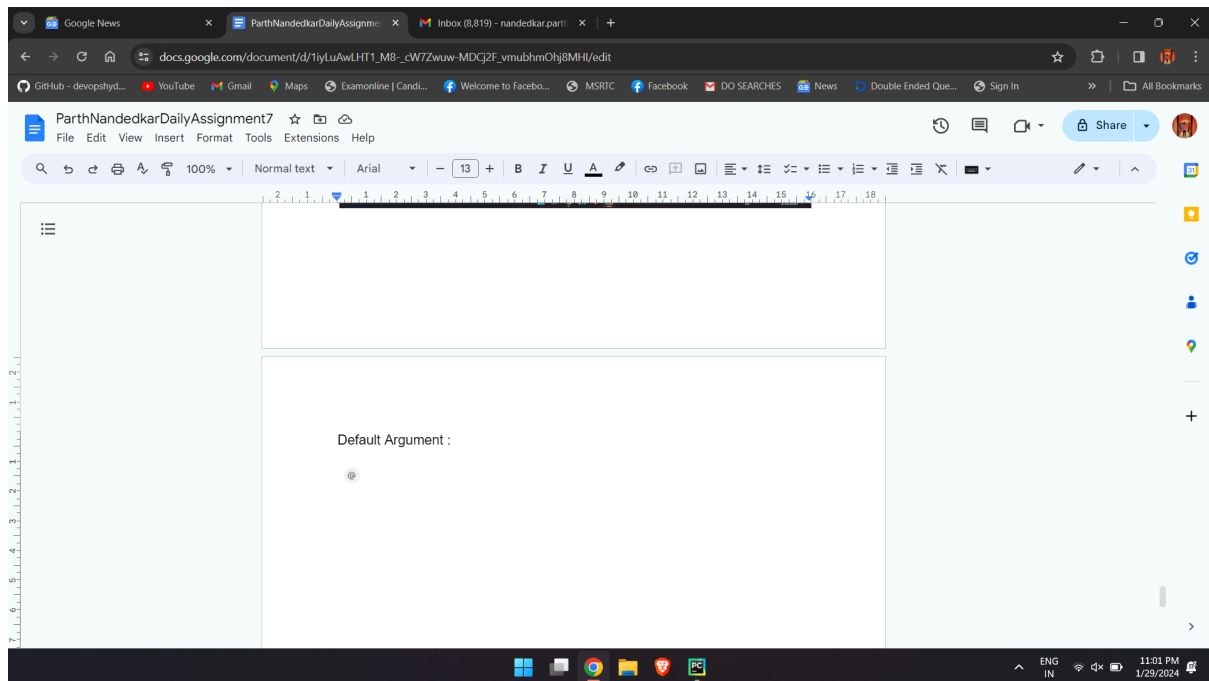


The screenshot shows the PyCharm IDE with the file `DateTimeFunction.py` open. The code imports `datetime` and `timedelta` from the `datetime` module. It gets the current date and time using `datetime.now()`, adds two days using `timedelta(days=2)`, and prints both the current date and the future date. The Run console shows the same command as the first screenshot, and the output is `15`. The process finished with exit code 0.

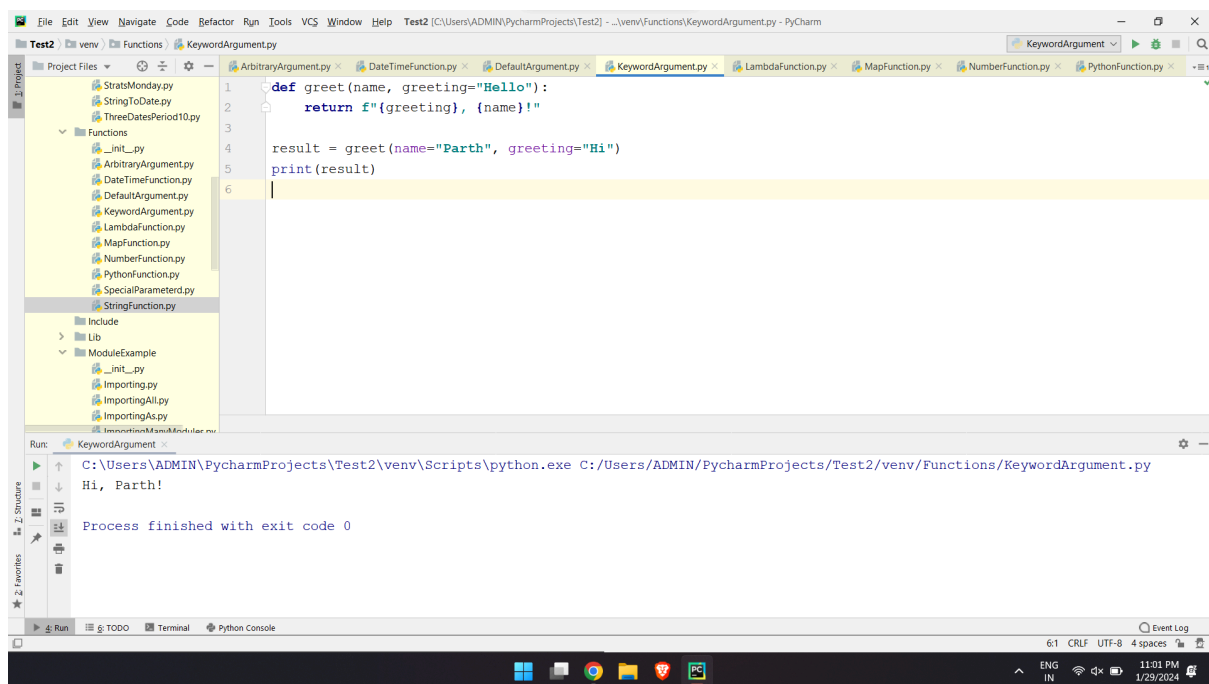
```
1 from datetime import datetime, timedelta
2
3 now = datetime.now()
4 future_date = now + timedelta(days=2)
5
6 print(now)
7 print(future_date)
8
```

Run: C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Functions/ArbitraryArgument.py
15
Process finished with exit code 0

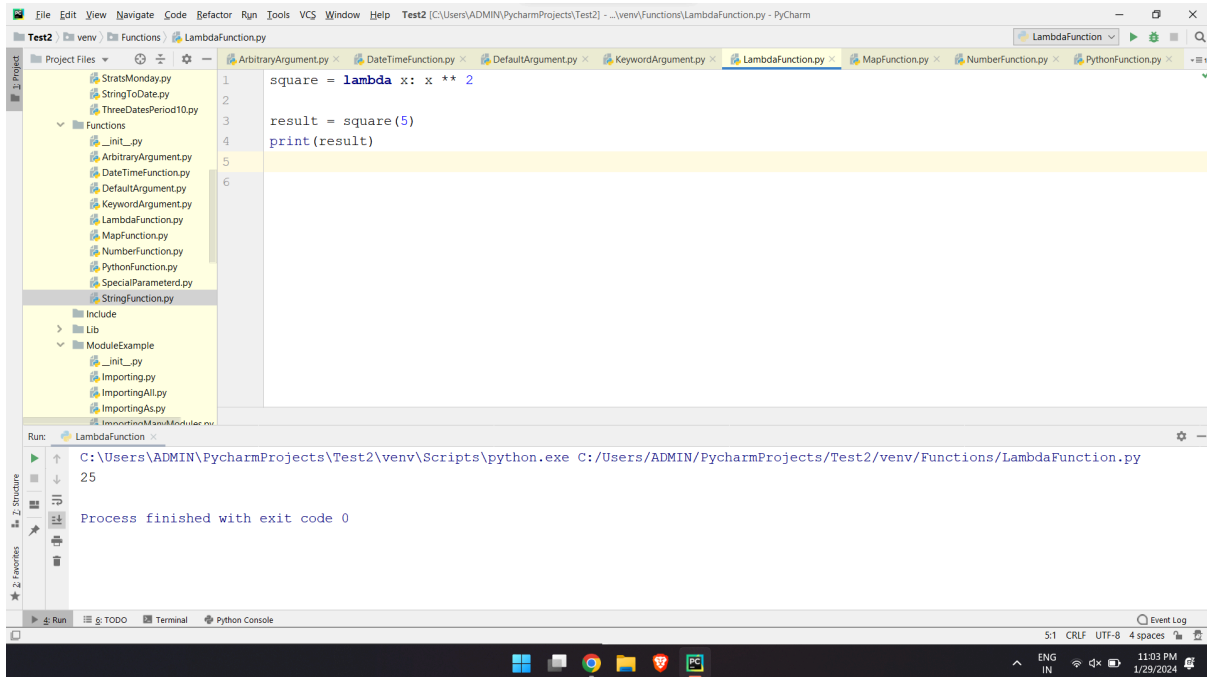
Default Argument :



Keyword argument :



Lambda function :



The screenshot shows the PyCharm IDE with a project named 'Test2'. The 'Functions' folder is expanded, showing various Python files. The 'LambdaFunction.py' file is open, displaying the following code:

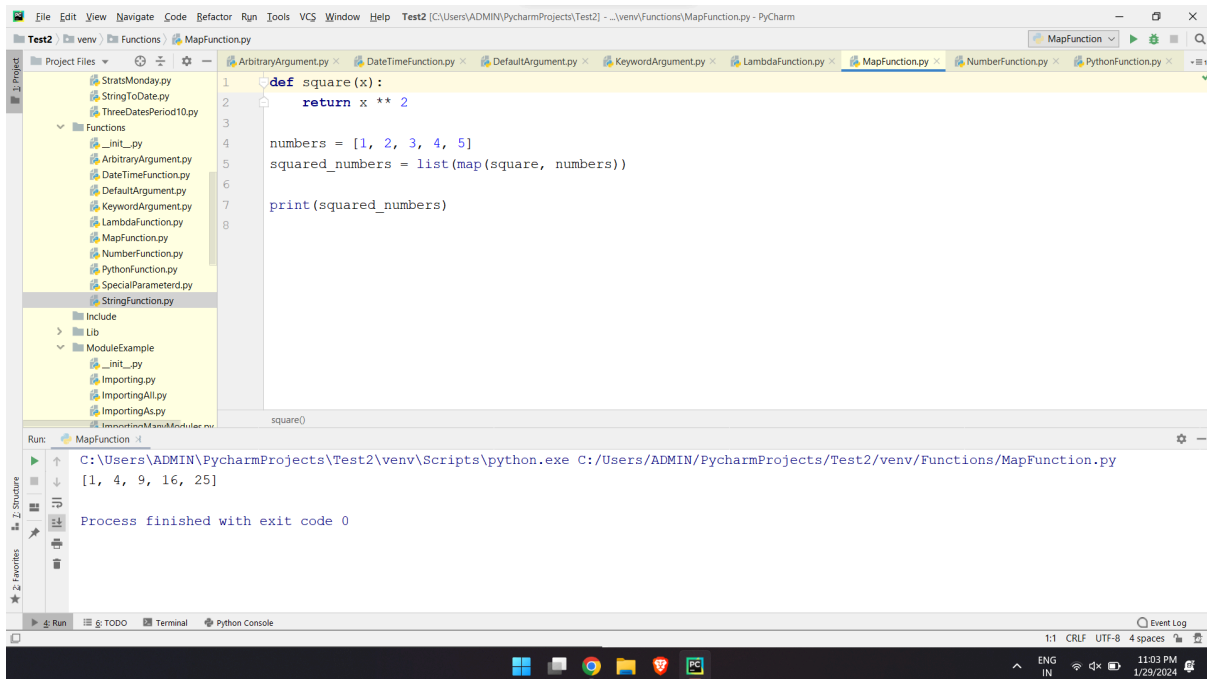
```
1 square = lambda x: x ** 2
2
3 result = square(5)
4 print(result)
5
6
```

The 'Run' button is clicked, and the output console shows the execution of the script:

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Functions/LambdaFunction.py
25
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8 and the line length is 51 characters.

Map function :



The screenshot shows the PyCharm IDE with a project named 'Test2'. The 'Functions' folder is expanded, showing various Python files. The 'MapFunction.py' file is open, displaying the following code:

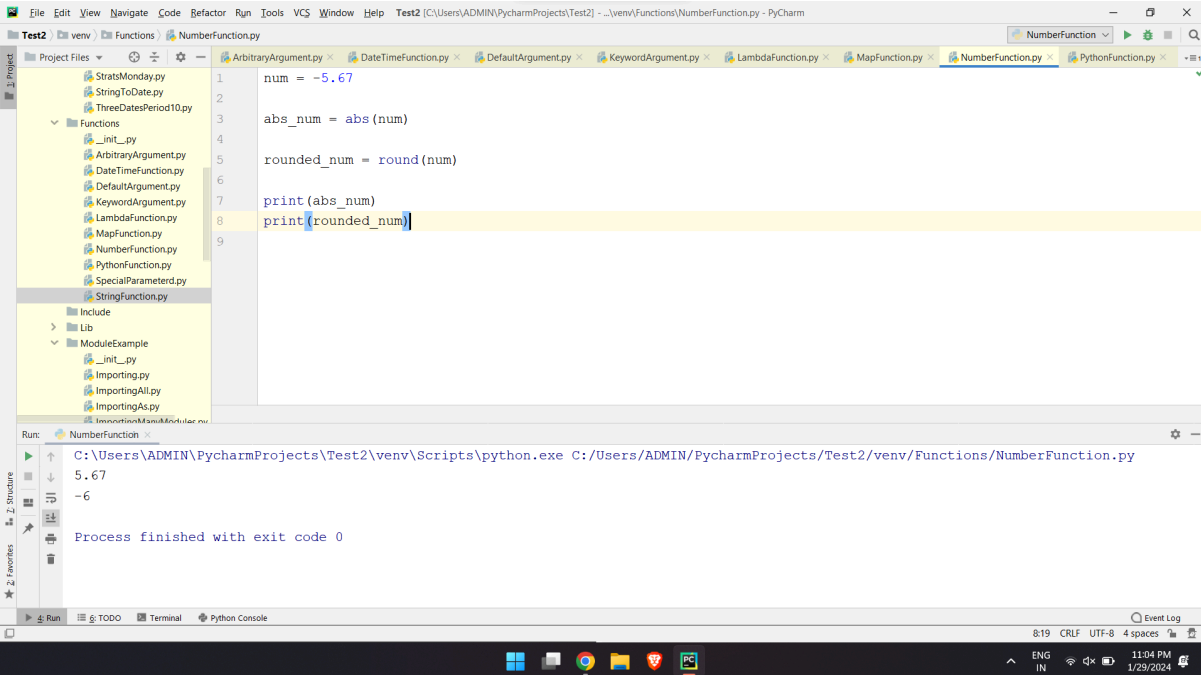
```
1 def square(x):
2     return x ** 2
3
4 numbers = [1, 2, 3, 4, 5]
5 squared_numbers = list(map(square, numbers))
6
7 print(squared_numbers)
8
```

The 'Run' button is clicked, and the output console shows the execution of the script:

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Functions/MapFunction.py
[1, 4, 9, 16, 25]
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8 and the line length is 11 characters.

Number function :

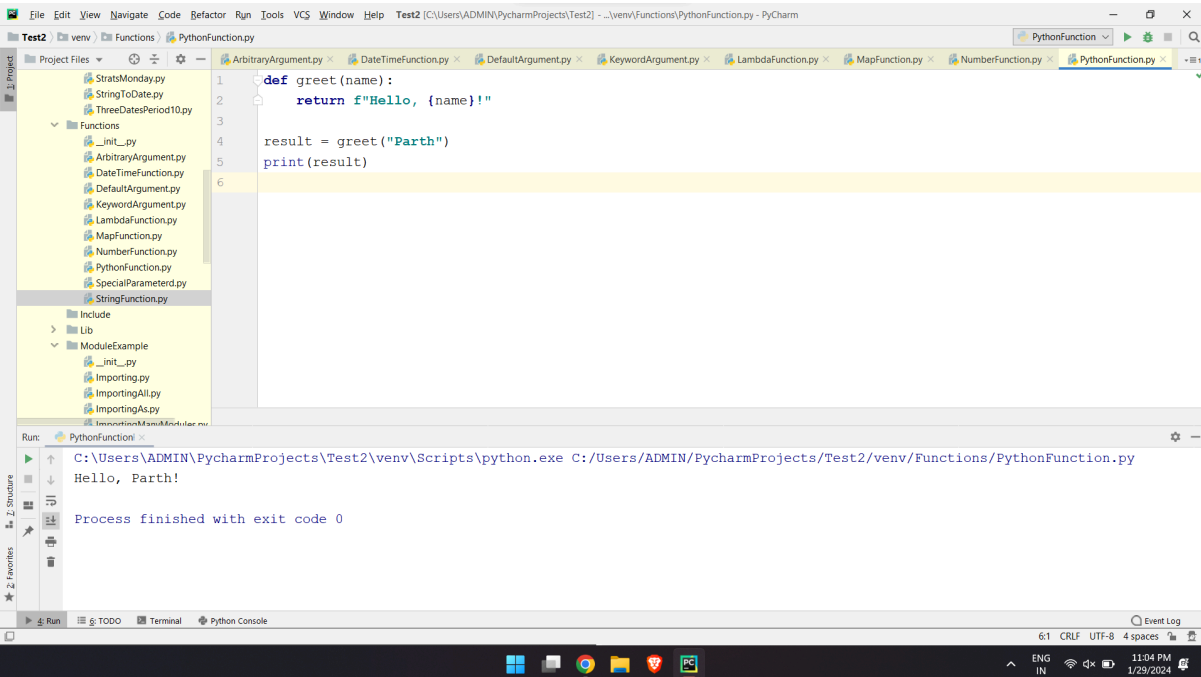


The screenshot shows the PyCharm IDE with the 'NumberFunction.py' file open. The code defines a variable 'num' with the value -5.67, calculates its absolute value 'abs_num', rounds it to 'rounded_num', and prints both. The Run console shows the output '5.67' and '-6', indicating a successful execution with exit code 0.

```
1 num = -5.67
2
3 abs_num = abs(num)
4
5 rounded_num = round(num)
6
7 print(abs_num)
8 print(rounded_num)
```

Run: NumberFunction
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Functions/NumberFunction.py
5.67
-6
Process finished with exit code 0

Python Functions :



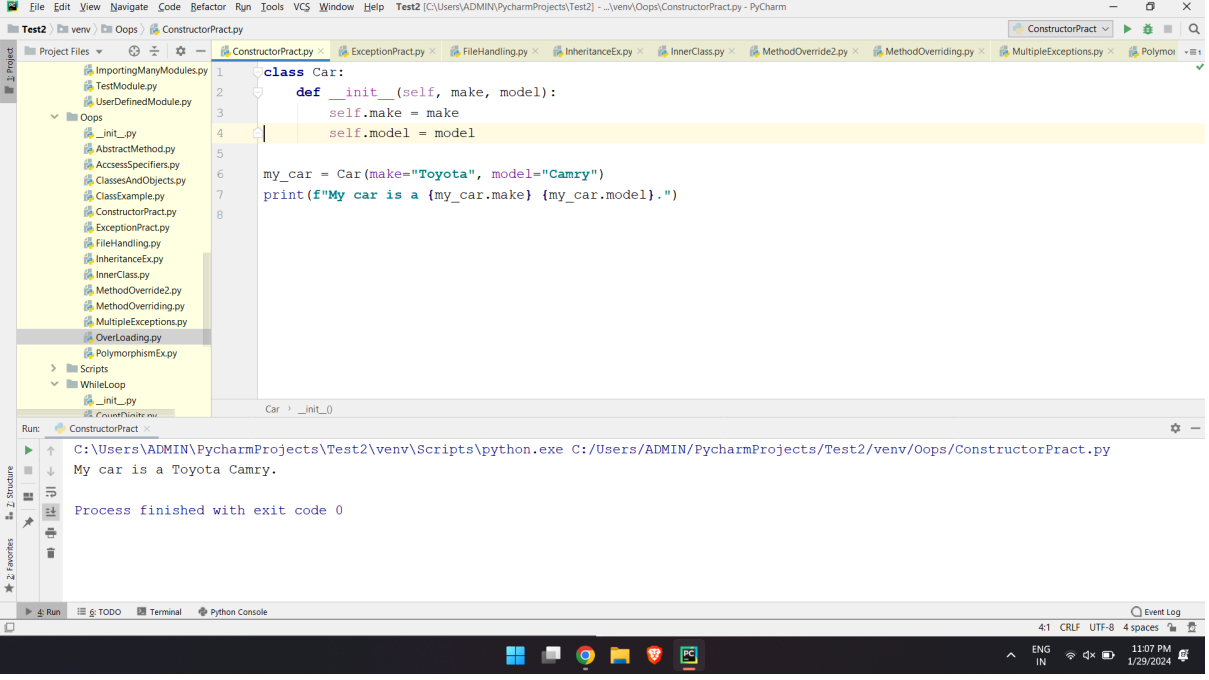
The screenshot shows the PyCharm IDE with the 'PythonFunction.py' file open. The code defines a 'greet' function that takes a name and returns a formatted string. It then calls the function with 'Parth' and prints the result. The Run console shows the output 'Hello, Parth!', indicating a successful execution with exit code 0.

```
1 def greet(name):
2     return f"Hello, {name}!"
3
4 result = greet("Parth")
5 print(result)
```

Run: PythonFunction
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Functions/PythonFunction.py
Hello, Parth!
Process finished with exit code 0

OOPS :

Constructor :

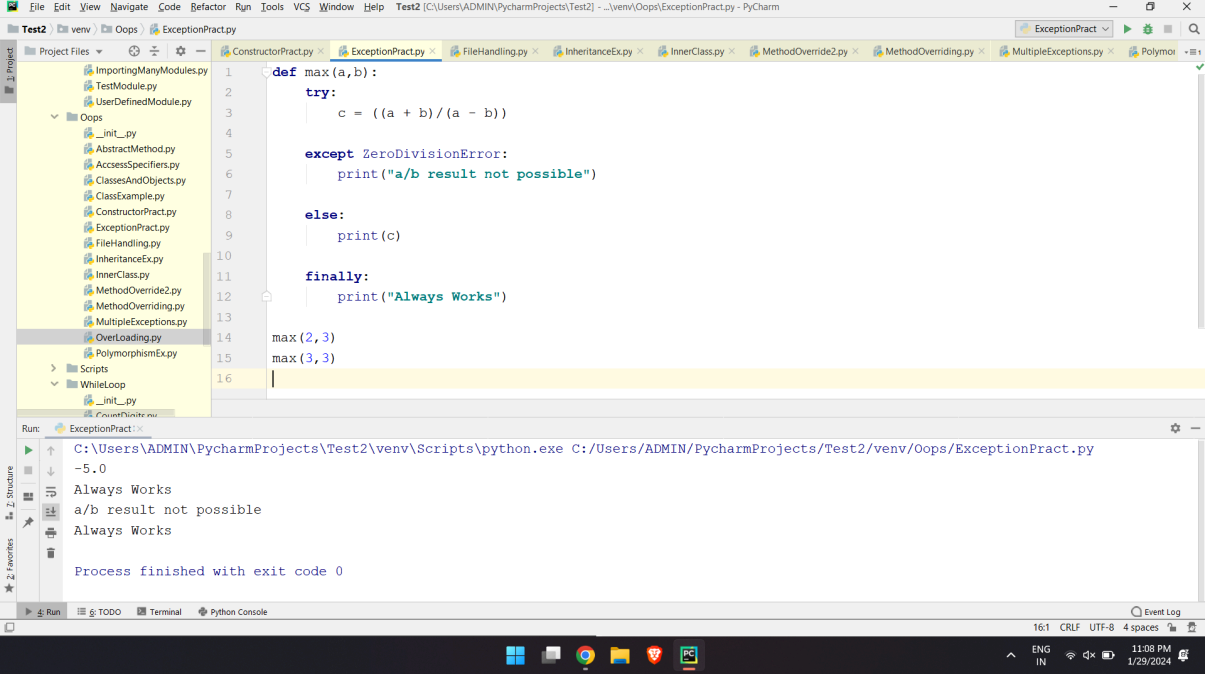


The screenshot shows the PyCharm IDE with the file `ConstructorPract.py` open. The code defines a `Car` class with an `__init__` method that takes `make` and `model` as arguments. An instance `my_car` is created with `make="Toyota"` and `model="Camry"`, and a print statement displays its attributes.

```
1 class Car:
2     def __init__(self, make, model):
3         self.make = make
4         self.model = model
5
6 my_car = Car(make="Toyota", model="Camry")
7 print(f"My car is a {my_car.make} {my_car.model}.")
8
```

The Run window shows the output: `My car is a Toyota Camry.` and `Process finished with exit code 0`.

Exception :

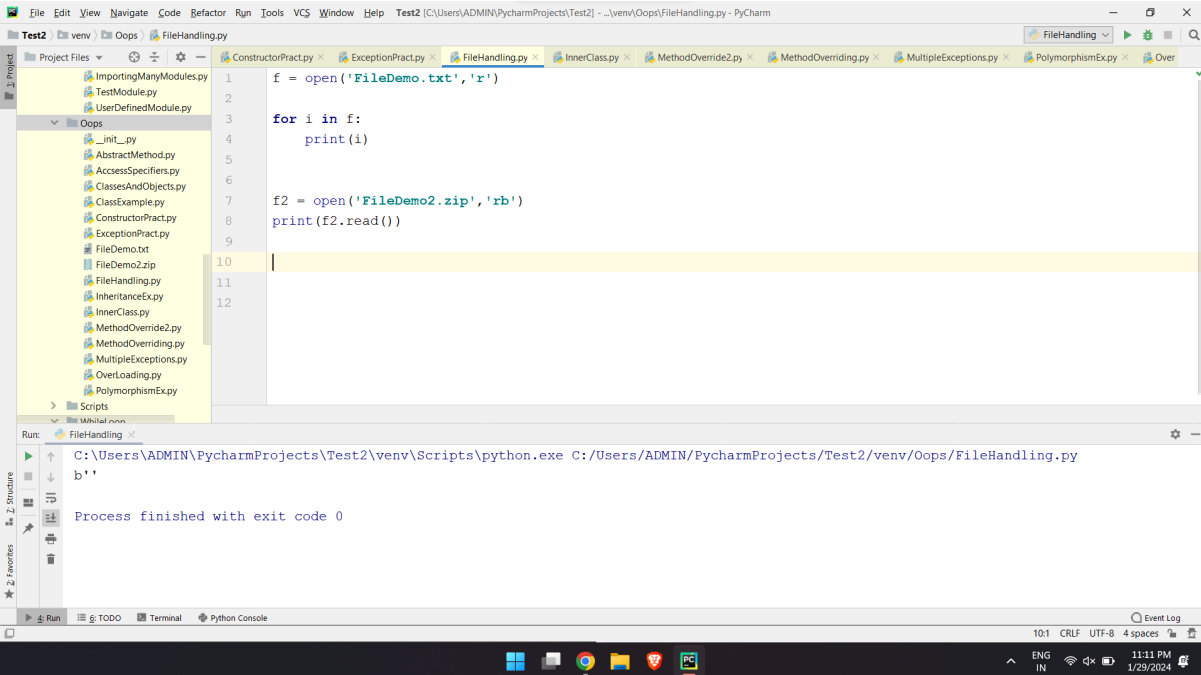


The screenshot shows the PyCharm IDE with the file `ExceptionPract.py` open. The code defines a `max` function that uses a `try-except-finally` block to handle a `ZeroDivisionError` when calculating `(a+b)/(a-b)`. It also prints "Always Works" in the `finally` block.

```
1 def max(a,b):
2     try:
3         c = ((a + b) / (a - b))
4
5     except ZeroDivisionError:
6         print("a/b result not possible")
7
8     else:
9         print(c)
10
11     finally:
12         print("Always Works")
13
14 max(2,3)
15 max(3,3)
16
```

The Run window shows the output: `-5.0`, `Always Works`, `a/b result not possible`, and `Always Works`. The process finished with exit code 0.

File Handling :

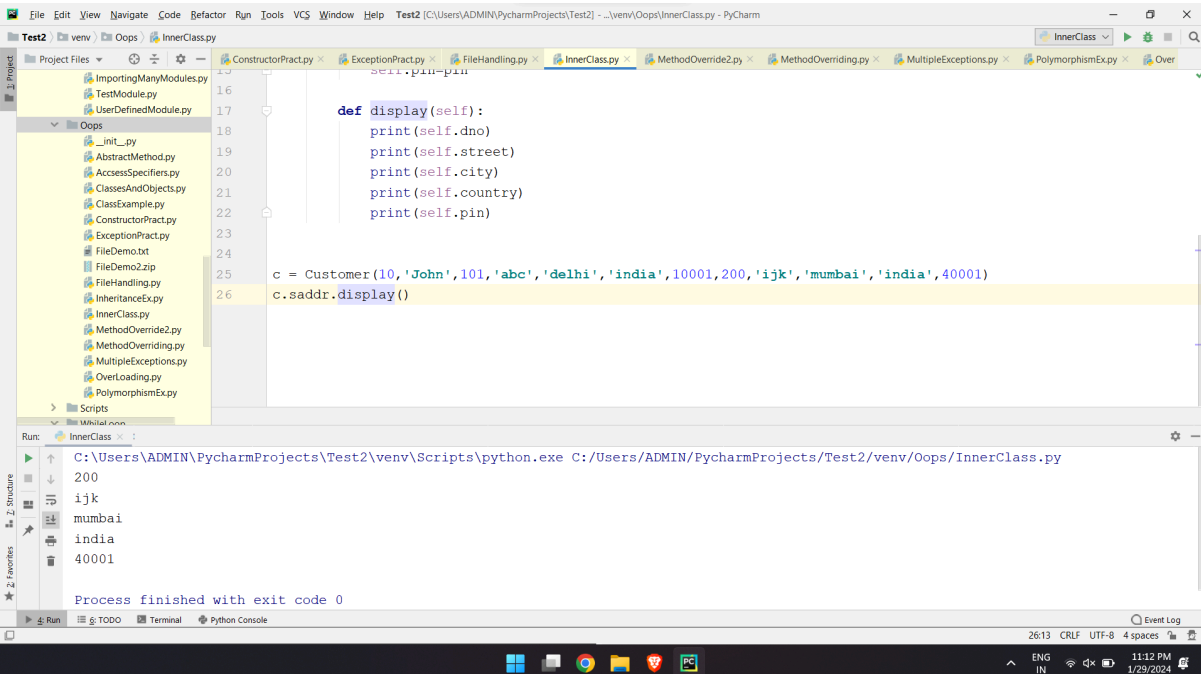


The screenshot shows the PyCharm IDE with a project named 'Test2'. The 'Project Files' pane on the left shows a directory structure with a folder named 'Oops' containing various Python files. The main editor displays the code in 'FileHandling.py':

```
1 f = open('FileDemo.txt', 'r')
2
3 for i in f:
4     print(i)
5
6
7 f2 = open('FileDemo2.zip', 'rb')
8 print(f2.read())
9
10
11
12
```

The 'Run' pane at the bottom shows the command executed: `C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Oops/FileHandling.py`. The output is `b''`, and the process finished with exit code 0.

Inner class :



The screenshot shows the PyCharm IDE with a project named 'Test2'. The 'Project Files' pane on the left shows a directory structure with a folder named 'Oops' containing various Python files. The main editor displays the code in 'InnerClass.py':

```
16
17
18
19
20
21
22
23
24
25
26
```

```
def display(self):
    print(self.dno)
    print(self.street)
    print(self.city)
    print(self.country)
    print(self.pin)

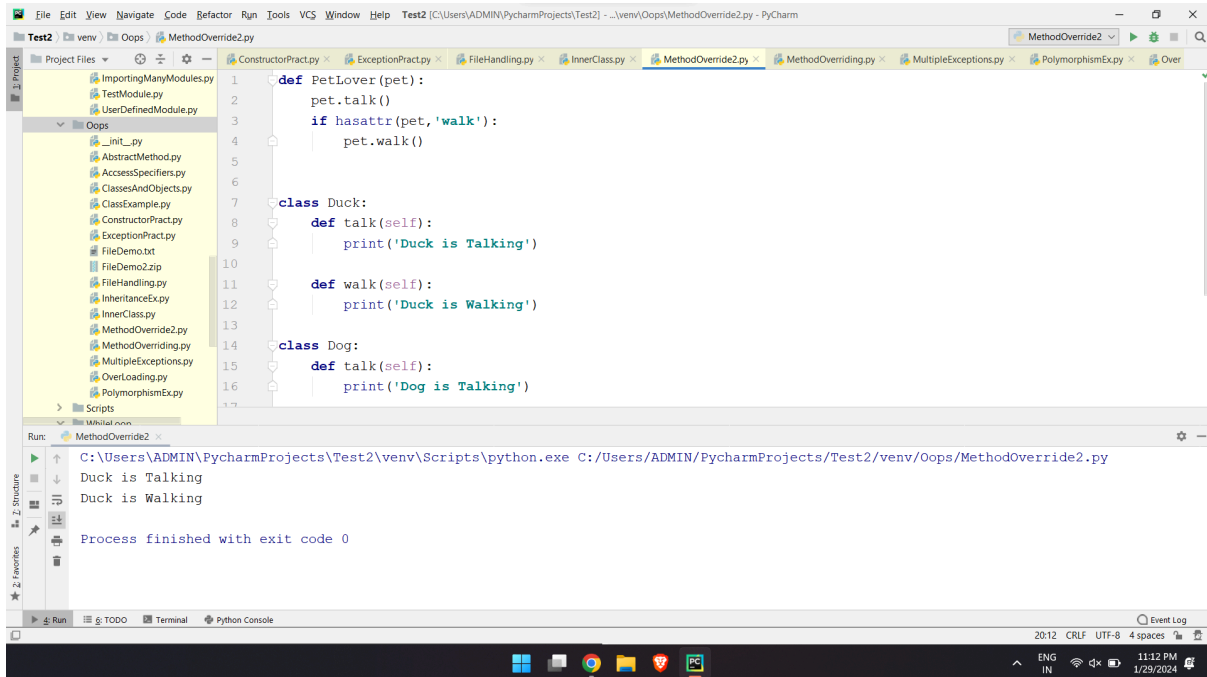
c = Customer(10, 'John', 101, 'abc', 'delhi', 'india', 10001, 200, 'ijk', 'mumbai', 'india', 40001)
c.saddr.display()
```

The 'Run' pane at the bottom shows the command executed: `C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Oops/InnerClass.py`. The output is:

```
200
ijk
mumbai
india
40001
```

The process finished with exit code 0.

Method Overriding :



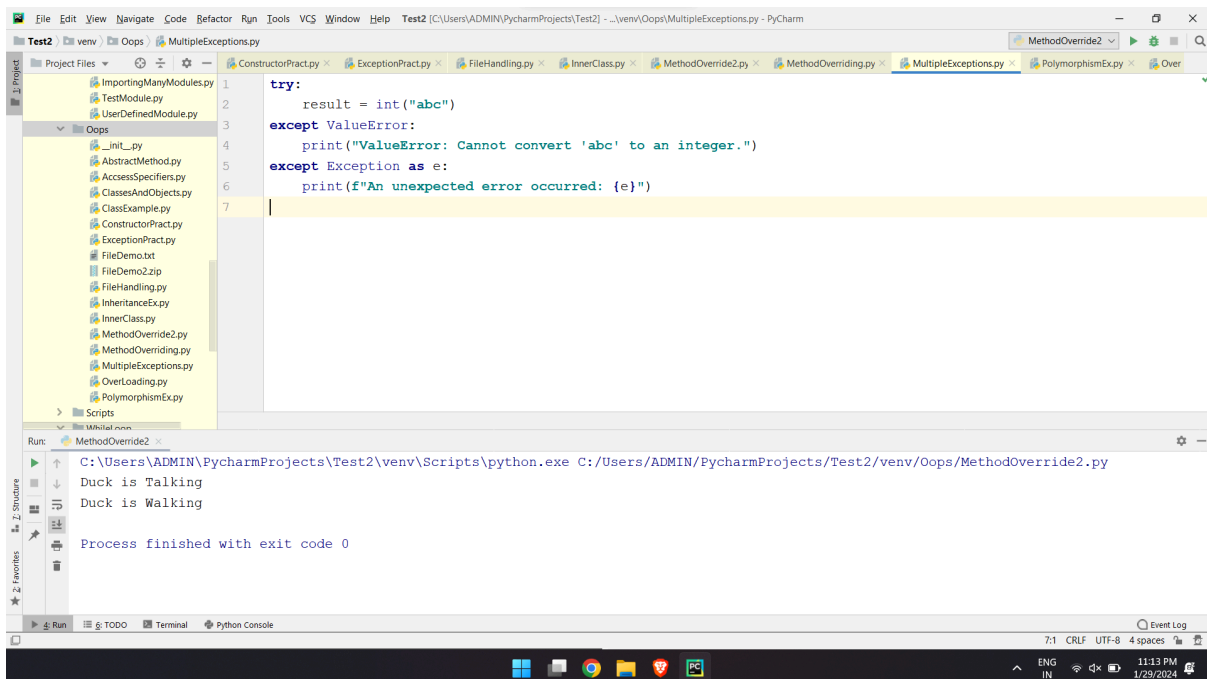
The screenshot shows the PyCharm IDE with a project named 'Test2'. The file explorer on the left shows a directory structure with 'Scripts' containing 'MethodOverride2.py'. The main editor displays the code for 'MethodOverride2.py'.

```
1 def PetLover(pet):
2     pet.talk()
3     if hasattr(pet, 'walk'):
4         pet.walk()
5
6
7 class Duck:
8     def talk(self):
9         print('Duck is Talking')
10
11     def walk(self):
12         print('Duck is Walking')
13
14 class Dog:
15     def talk(self):
16         print('Dog is Talking')
```

The Run window at the bottom shows the execution output:

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Oops/MethodOverride2.py
Duck is Talking
Duck is Walking
Process finished with exit code 0
```

Multiple Exceptions :



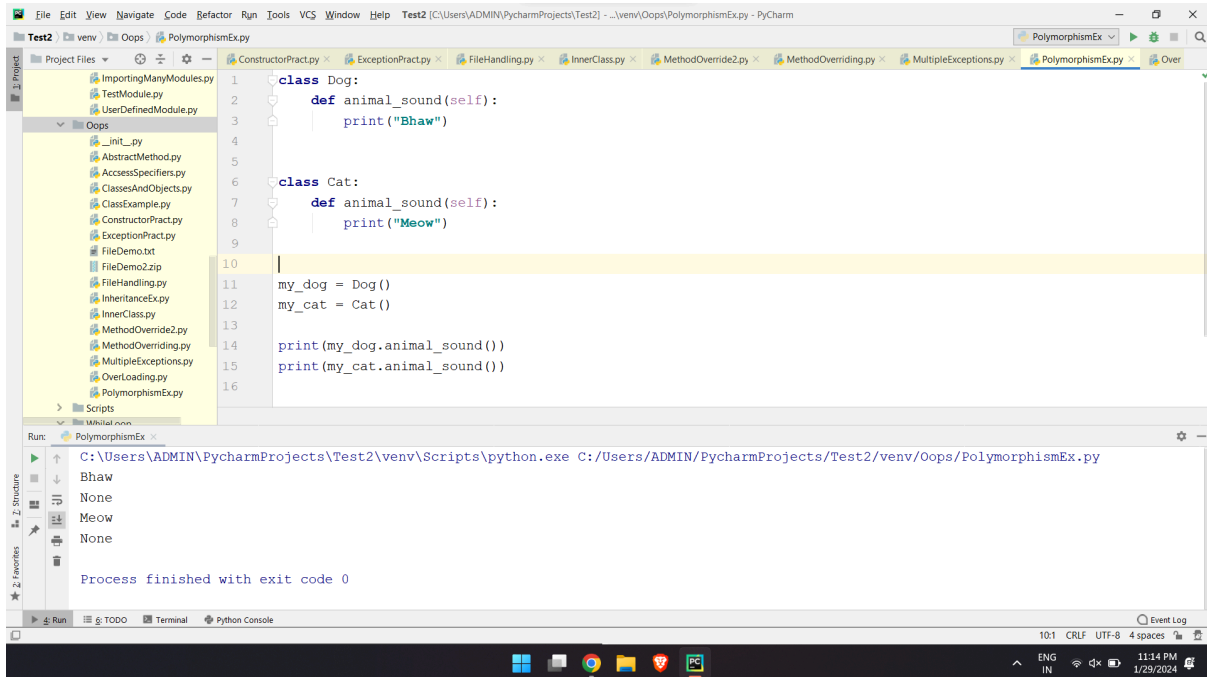
The screenshot shows the PyCharm IDE with a project named 'Test2'. The file explorer on the left shows a directory structure with 'Scripts' containing 'MultipleExceptions.py'. The main editor displays the code for 'MultipleExceptions.py'.

```
1 try:
2     result = int("abc")
3 except ValueError:
4     print("ValueError: Cannot convert 'abc' to an integer.")
5 except Exception as e:
6     print(f"An unexpected error occurred: {e}")
7
```

The Run window at the bottom shows the execution output:

```
C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Oops/MethodOverride2.py
Duck is Talking
Duck is Walking
Process finished with exit code 0
```

Polymorphism :



The screenshot shows a PyCharm IDE window titled 'Test2' with the file 'PolymorphismEx.py' open. The code defines two classes, 'Dog' and 'Cat', each with an 'animal_sound' method. The 'Dog' method prints 'Bhaw' and the 'Cat' method prints 'Meow'. Below the class definitions, the code creates instances 'my_dog' and 'my_cat', and then calls their 'animal_sound' methods. The output window shows the execution results: 'Bhaw' and 'Meow' on separate lines, followed by 'Process finished with exit code 0'.

```
1 class Dog:
2     def animal_sound(self):
3         print("Bhaw")
4
5
6 class Cat:
7     def animal_sound(self):
8         print("Meow")
9
10
11 my_dog = Dog()
12 my_cat = Cat()
13
14 print(my_dog.animal_sound())
15 print(my_cat.animal_sound())
16
```

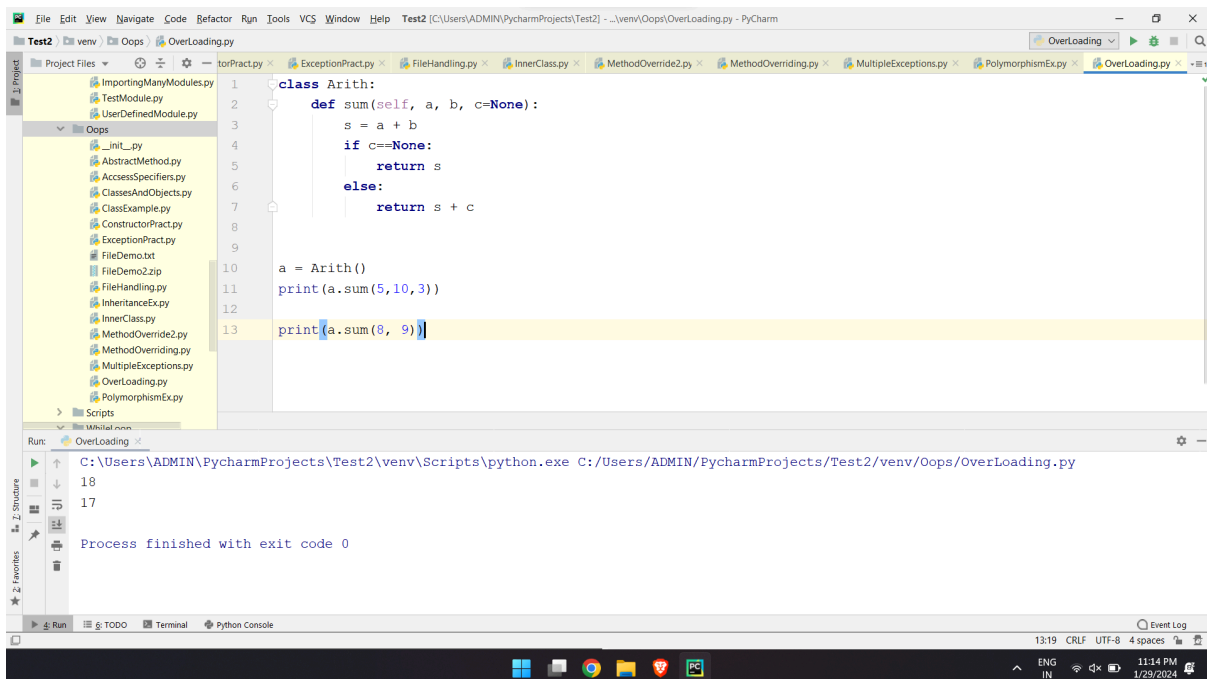
Run: PolymorphismEx

C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Oops/PolymorphismEx.py

Bhaw
Meow
None
None

Process finished with exit code 0

OverLoading :



The screenshot shows a PyCharm IDE window titled 'Test2' with the file 'OverLoading.py' open. The code defines a class 'Arith' with a 'sum' method that takes three arguments: 'a', 'b', and 'c'. The method uses a conditional return statement to handle different numbers of arguments. Below the class definition, the code creates an instance 'a' of the 'Arith' class and calls the 'sum' method with different sets of arguments: (5, 10, 3) and (8, 9). The output window shows the execution results: '18' and '17' on separate lines, followed by 'Process finished with exit code 0'.

```
1 class Arith:
2     def sum(self, a, b, c=None):
3         s = a + b
4         if c==None:
5             return s
6         else:
7             return s + c
8
9
10 a = Arith()
11 print(a.sum(5,10,3))
12
13 print(a.sum(8, 9))
```

Run: OverLoading

C:\Users\ADMIN\PycharmProjects\Test2\venv\Scripts\python.exe C:/Users/ADMIN/PycharmProjects/Test2/venv/Oops/OverLoading.py

18
17

Process finished with exit code 0