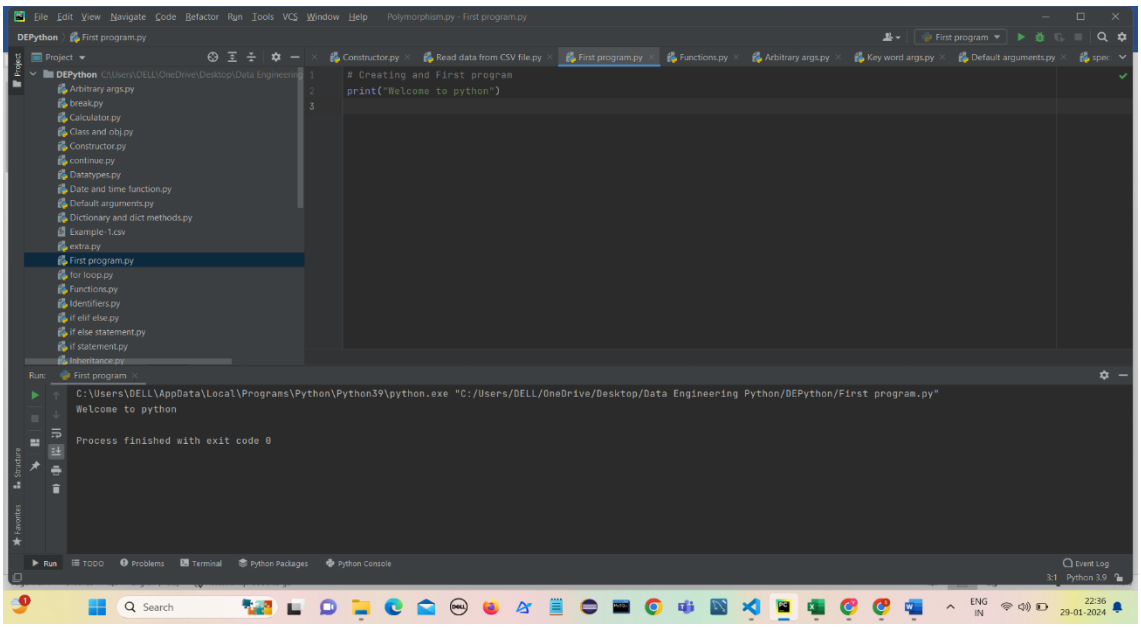
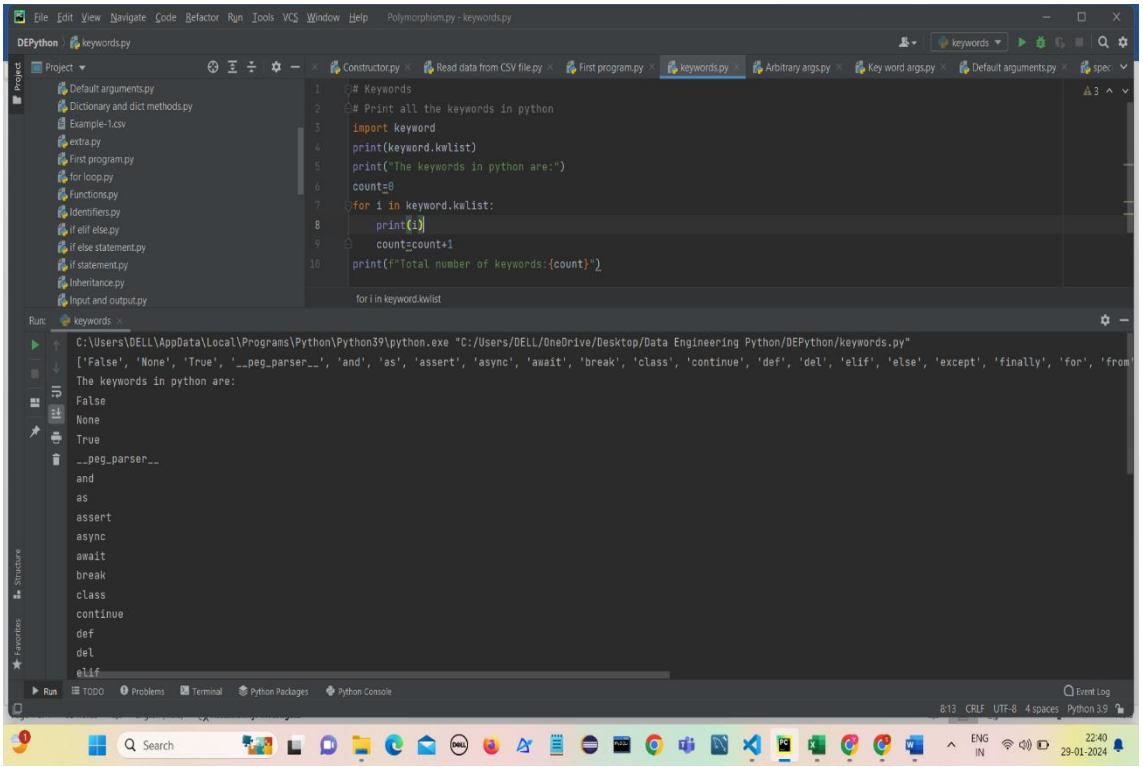
Python
Assignment-1

1) Writing and Running First Program



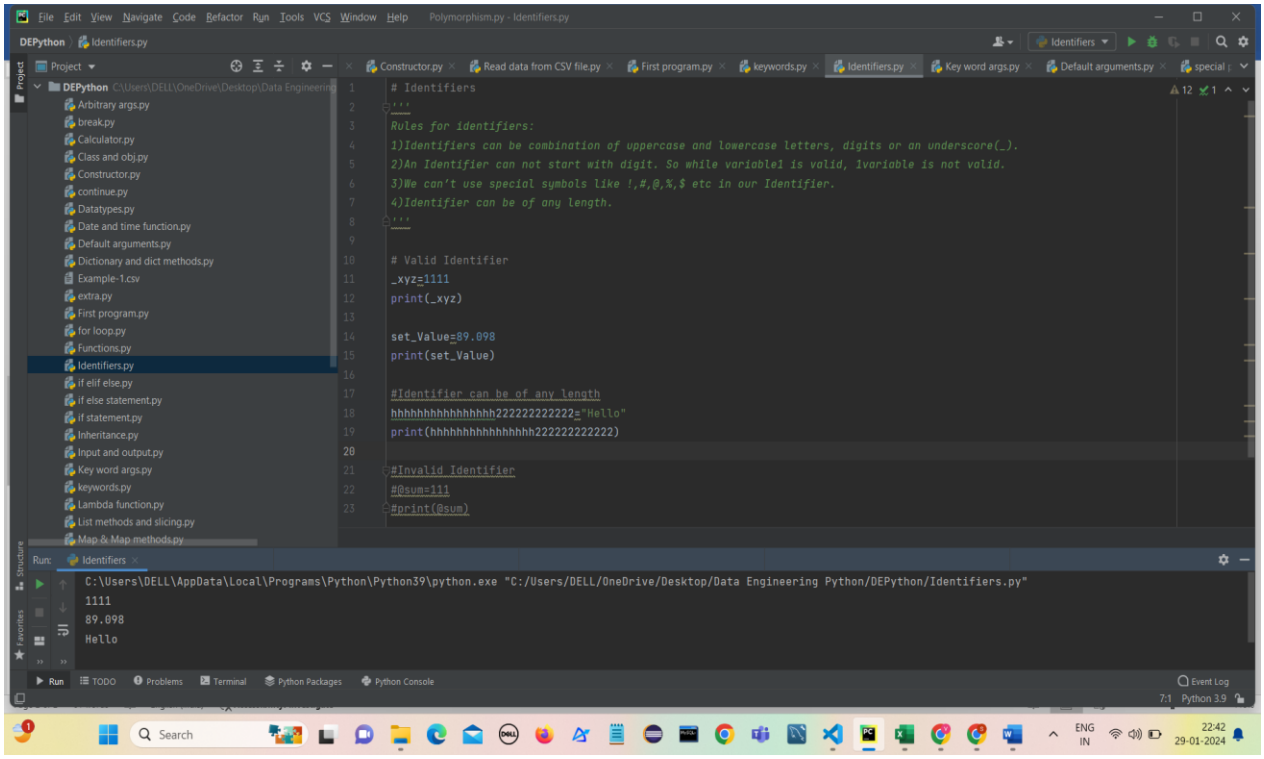
2) Key words

Keywords in Python are reserved words that can not be used as a variable name, function name, or any other identifier.



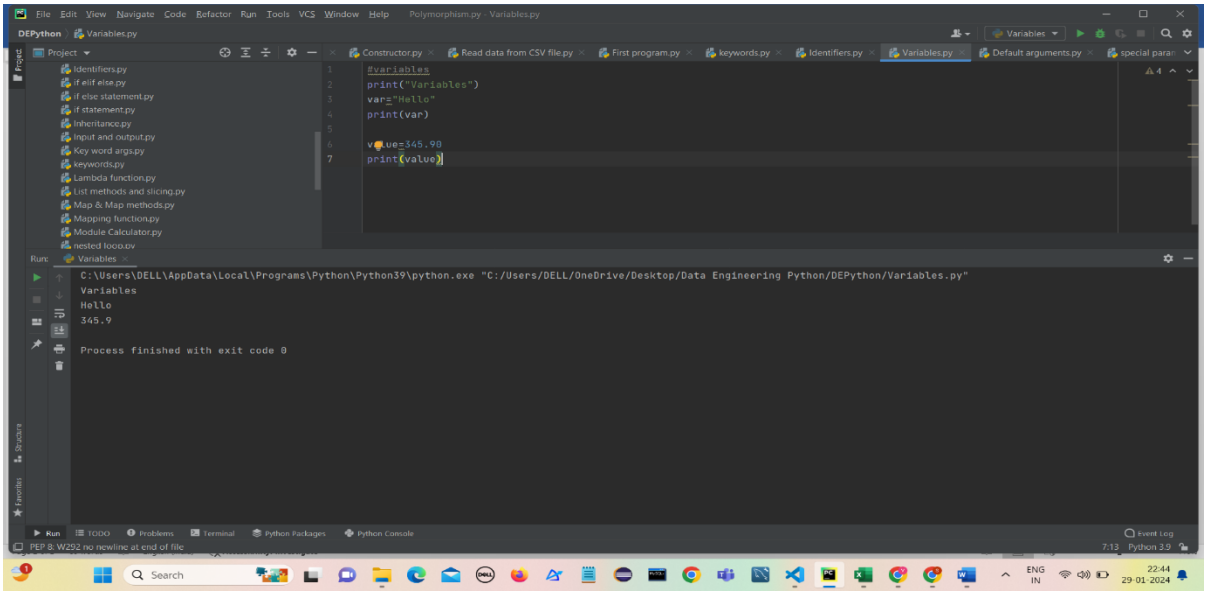
3) Identifiers

- **Identifier** is a user-defined name given to a variable, function, class, module, etc.
- The identifier is a combination of character digits and an underscore. They are case-sensitive.



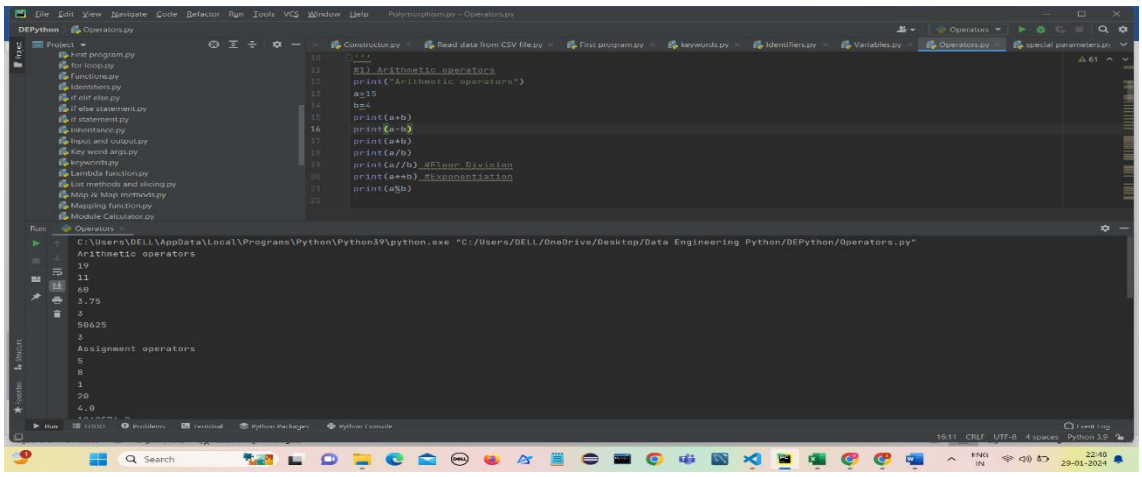
4) Variables

- Python Variable is containers that store values.
- Python is not “statically typed”.
- We do not need to declare variables before using them or declare their type.

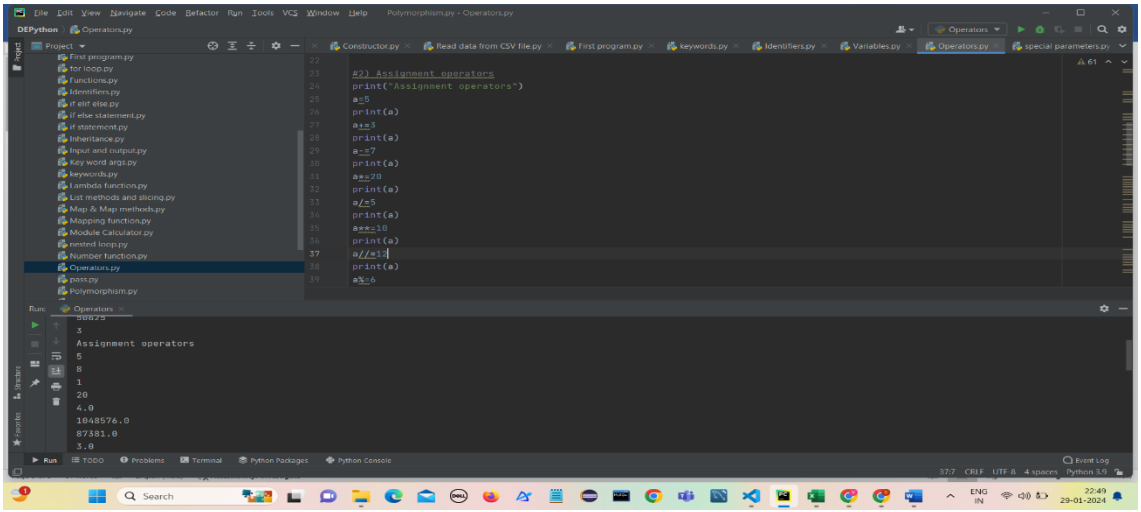


5) Operators

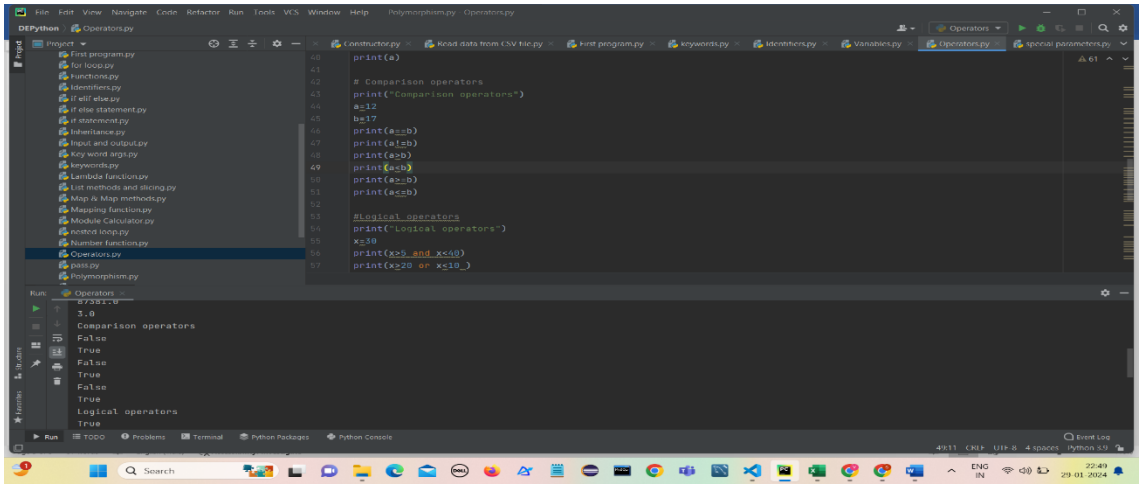
i) Arithmetic Operators



ii) Assignment Operators

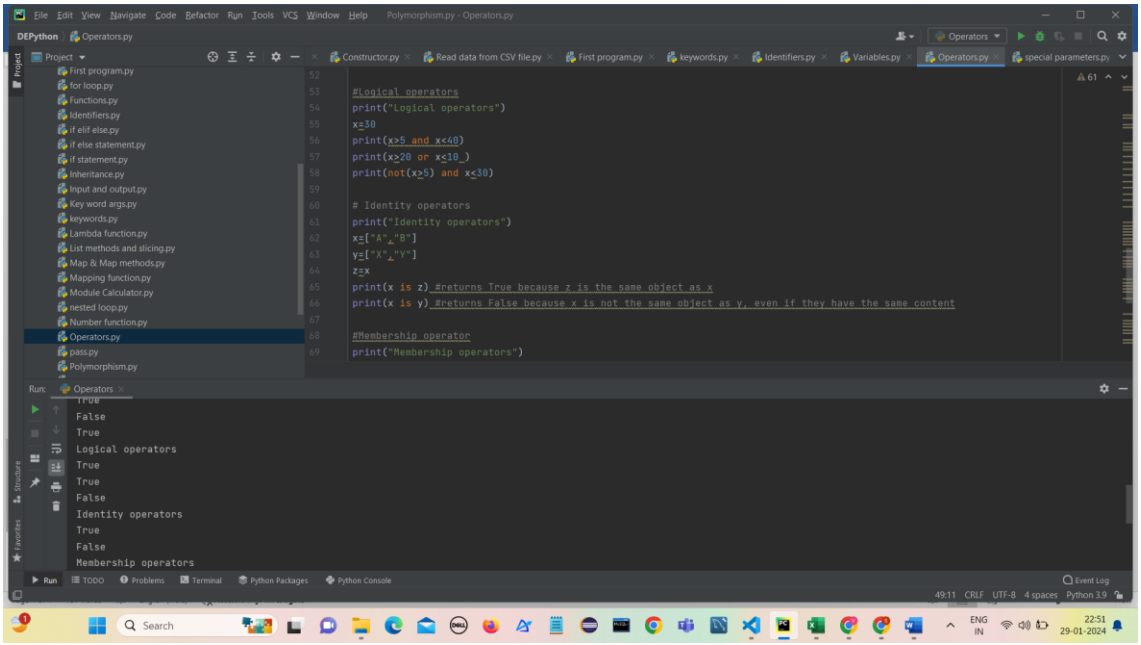


iii) Comparison Operators



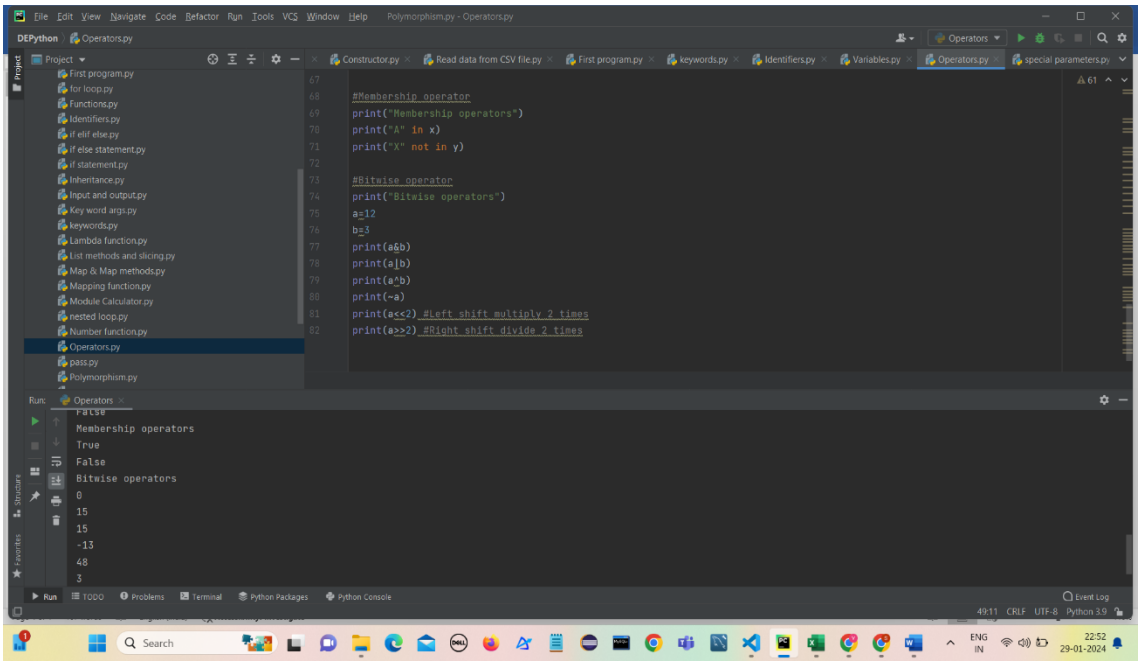
iv) Logical Operators

v) Identity Operators

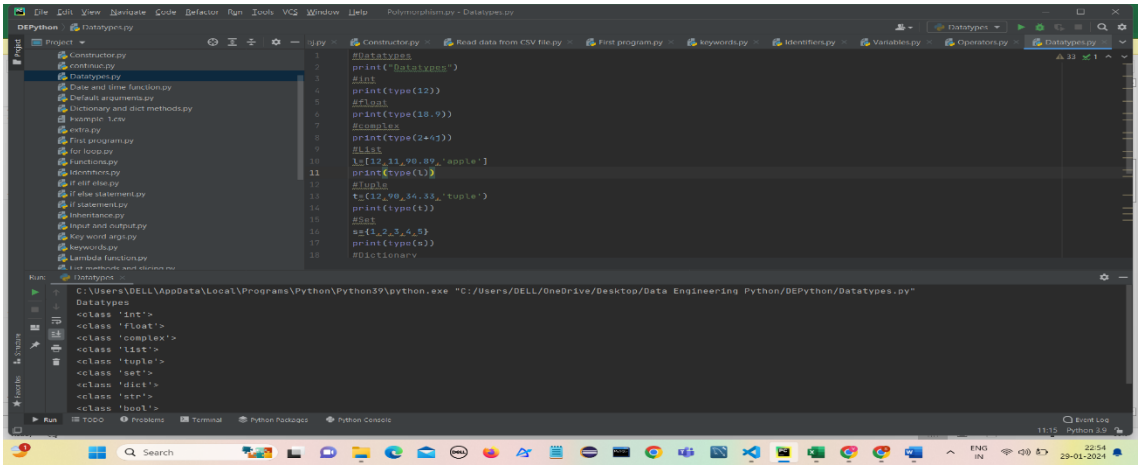


vi) Membership Operators

vii) Bitwise Operators

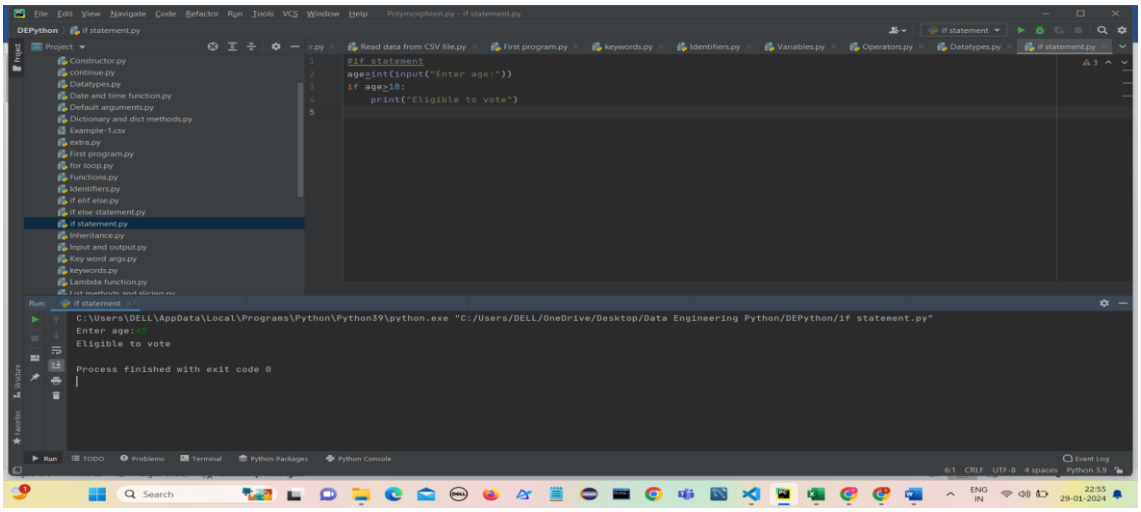


6) Datatypes

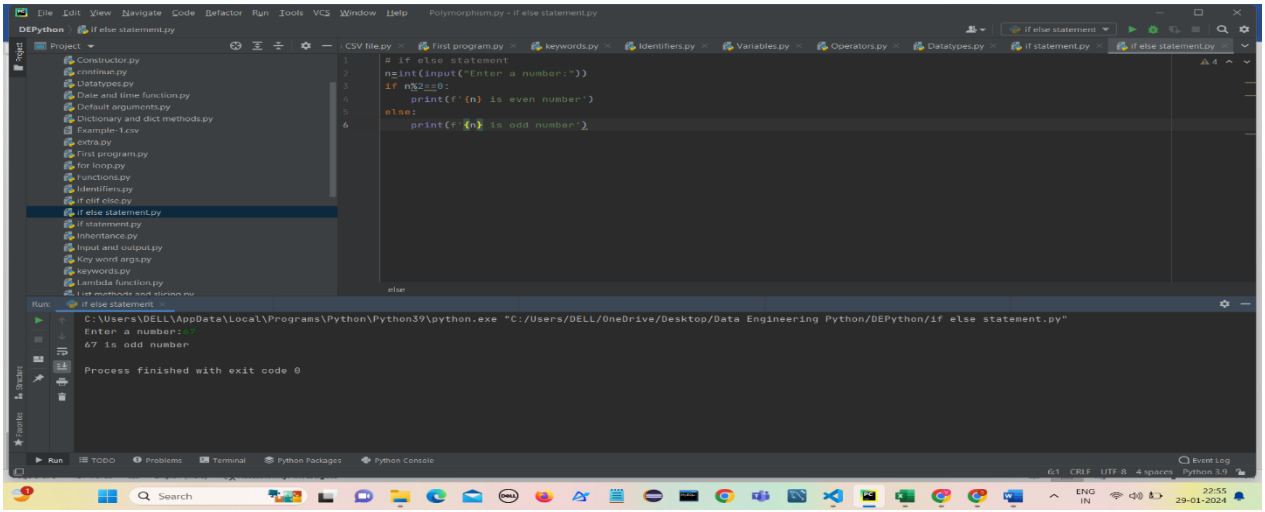


Control Structures

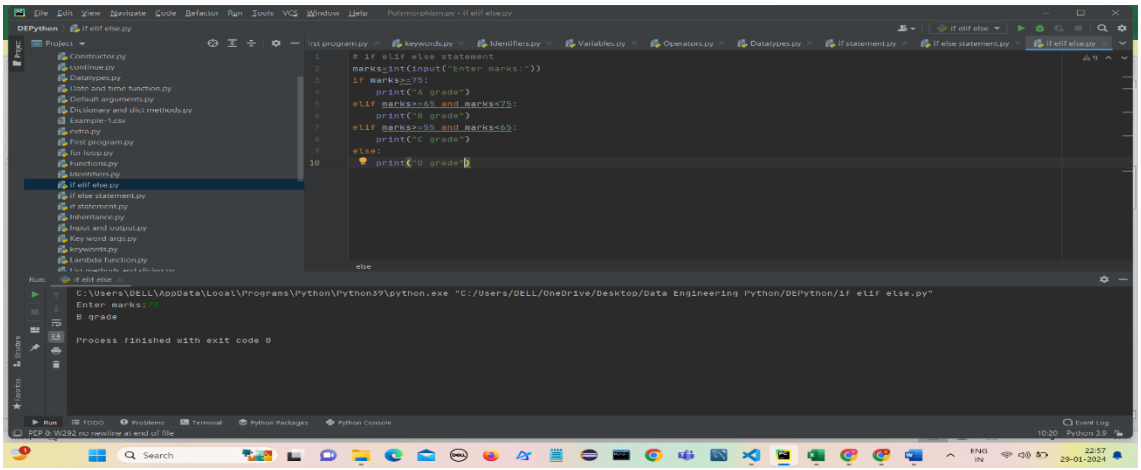
7) If statement



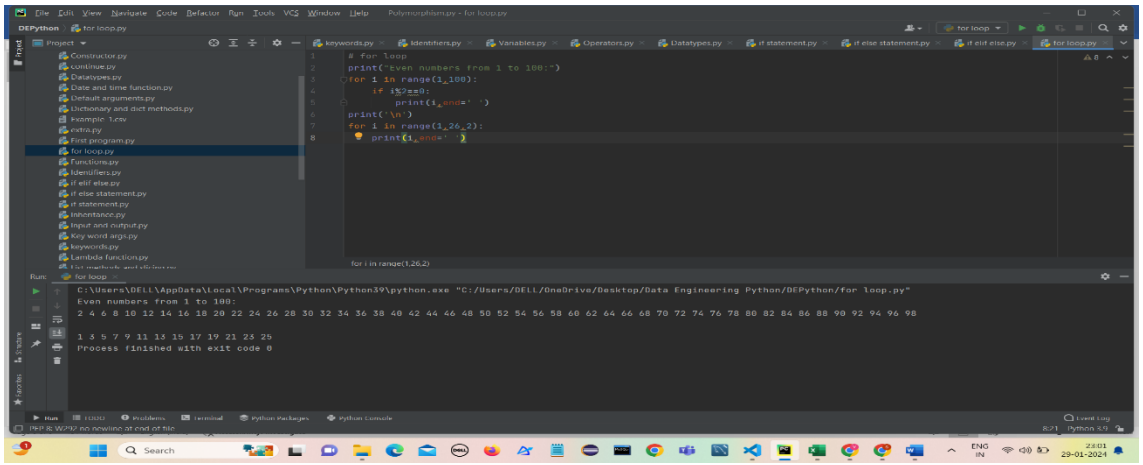
8) If else statement



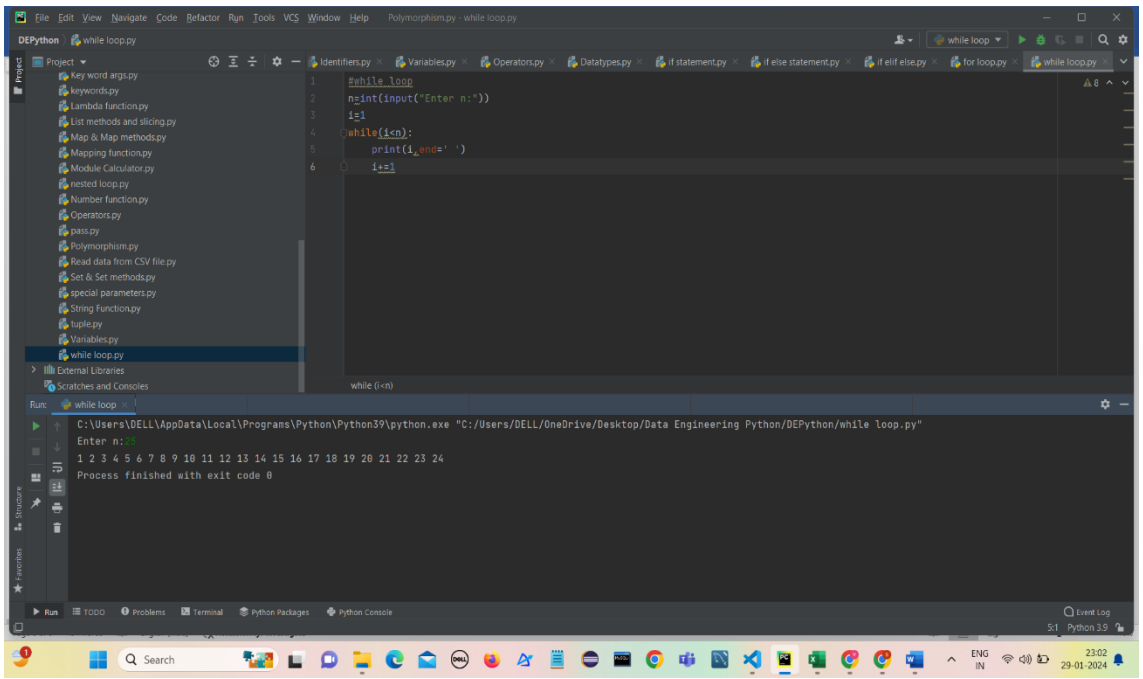
9) If elif else statement



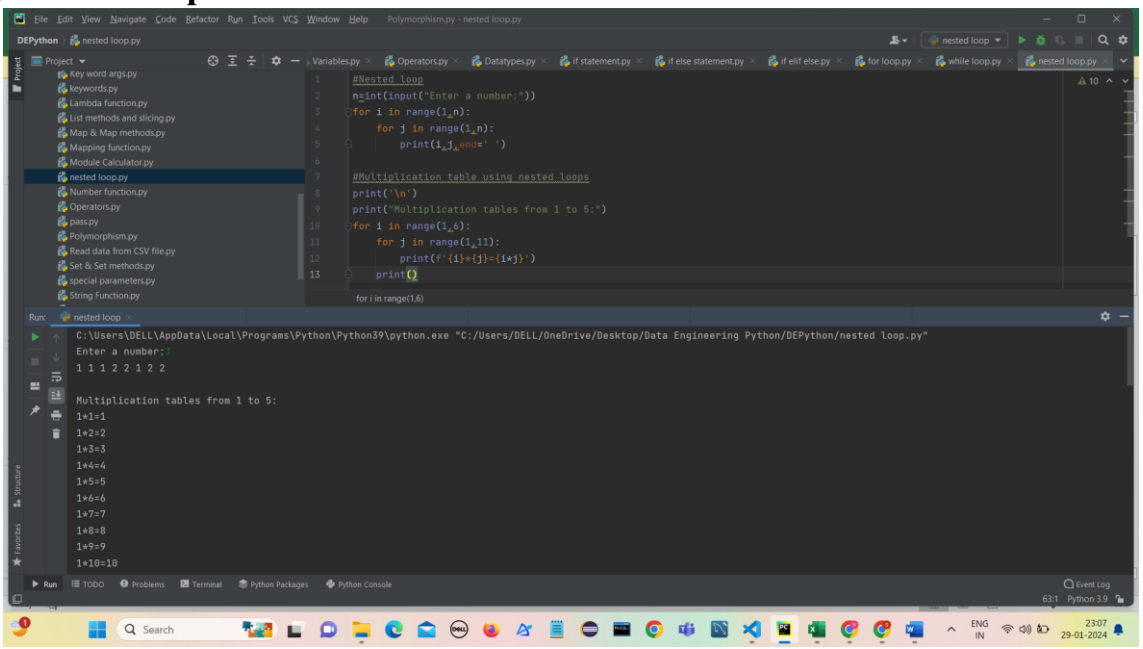
10) For Loop



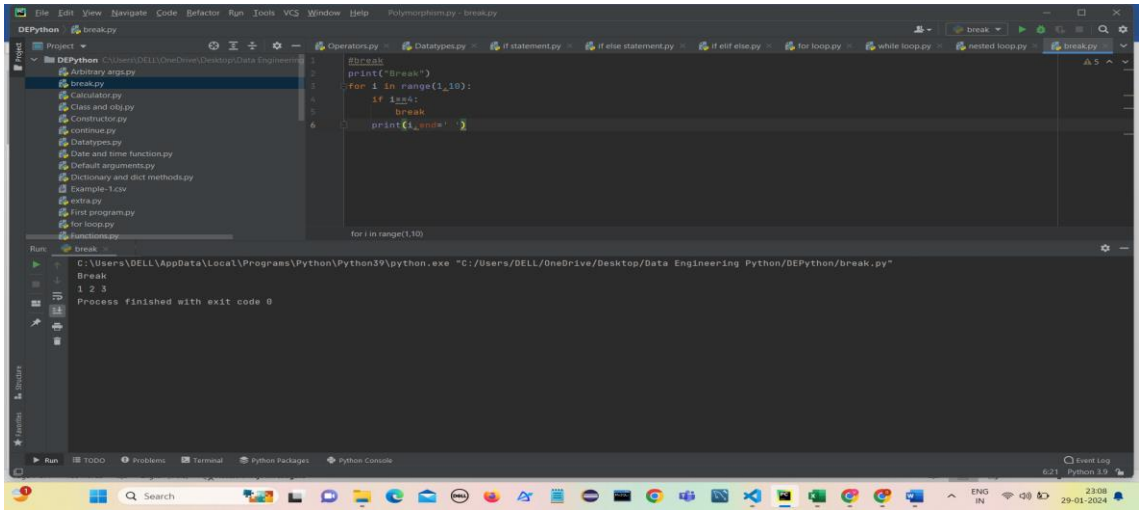
11) While loop



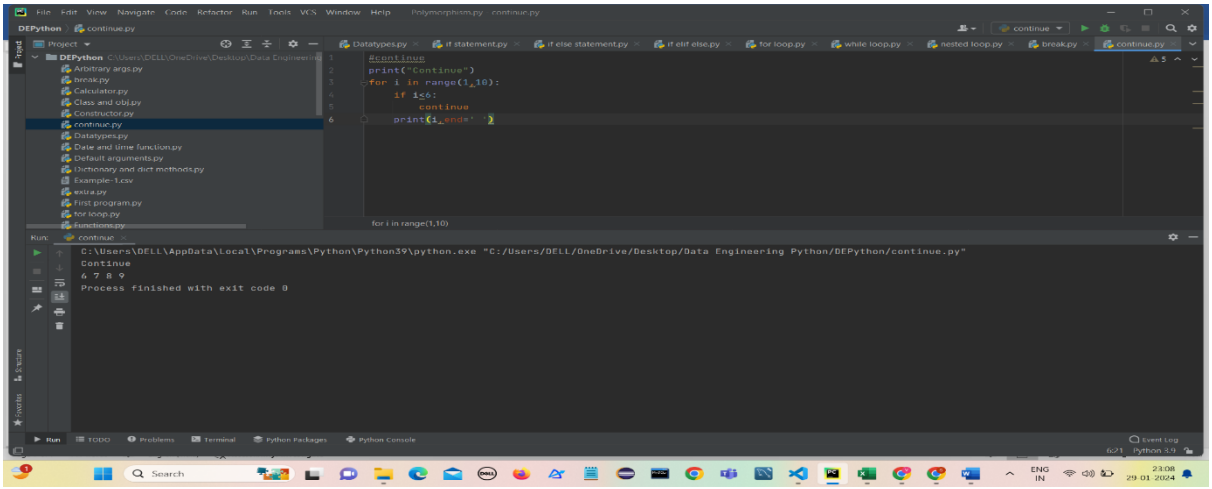
12) Nested loop



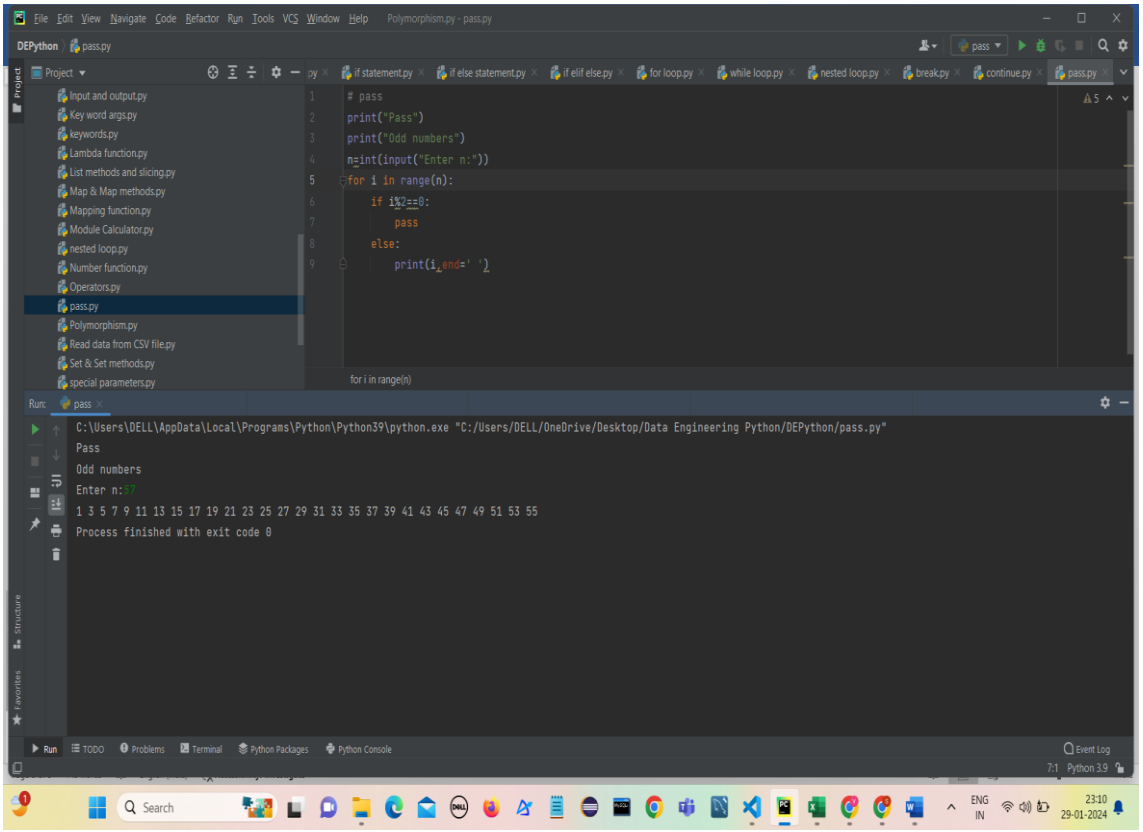
13) Break



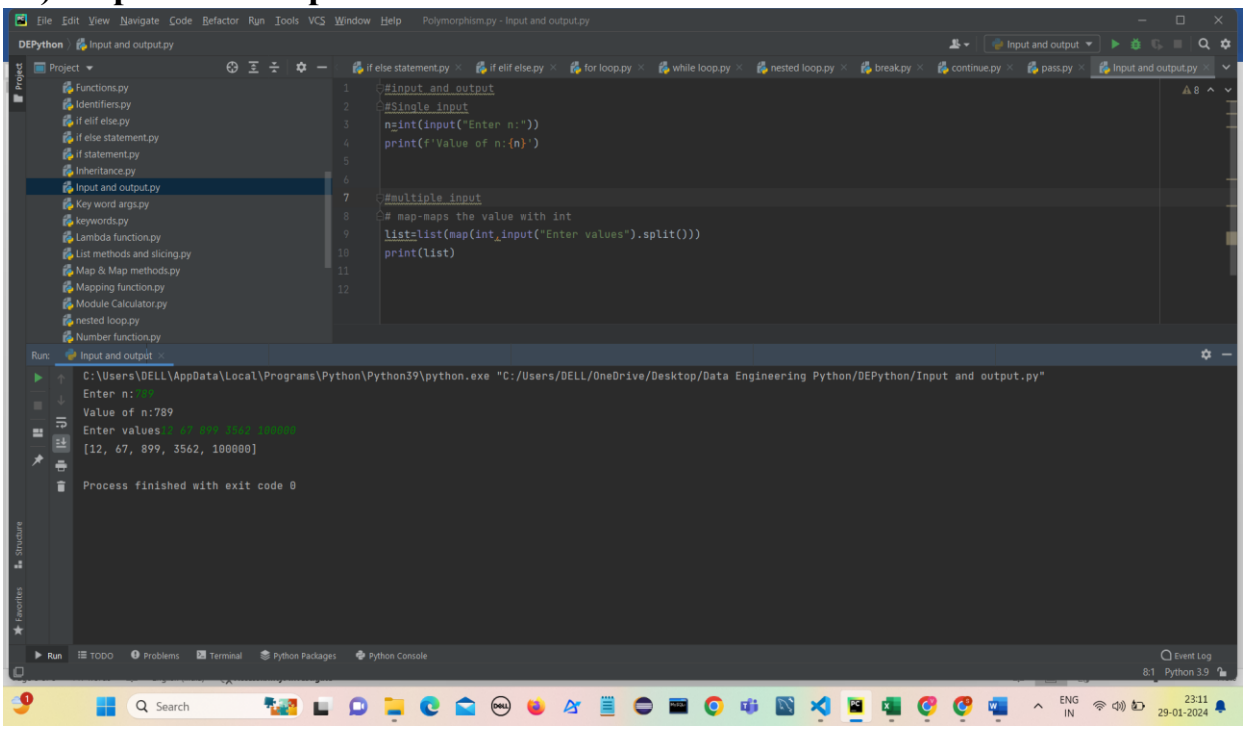
14) Continue



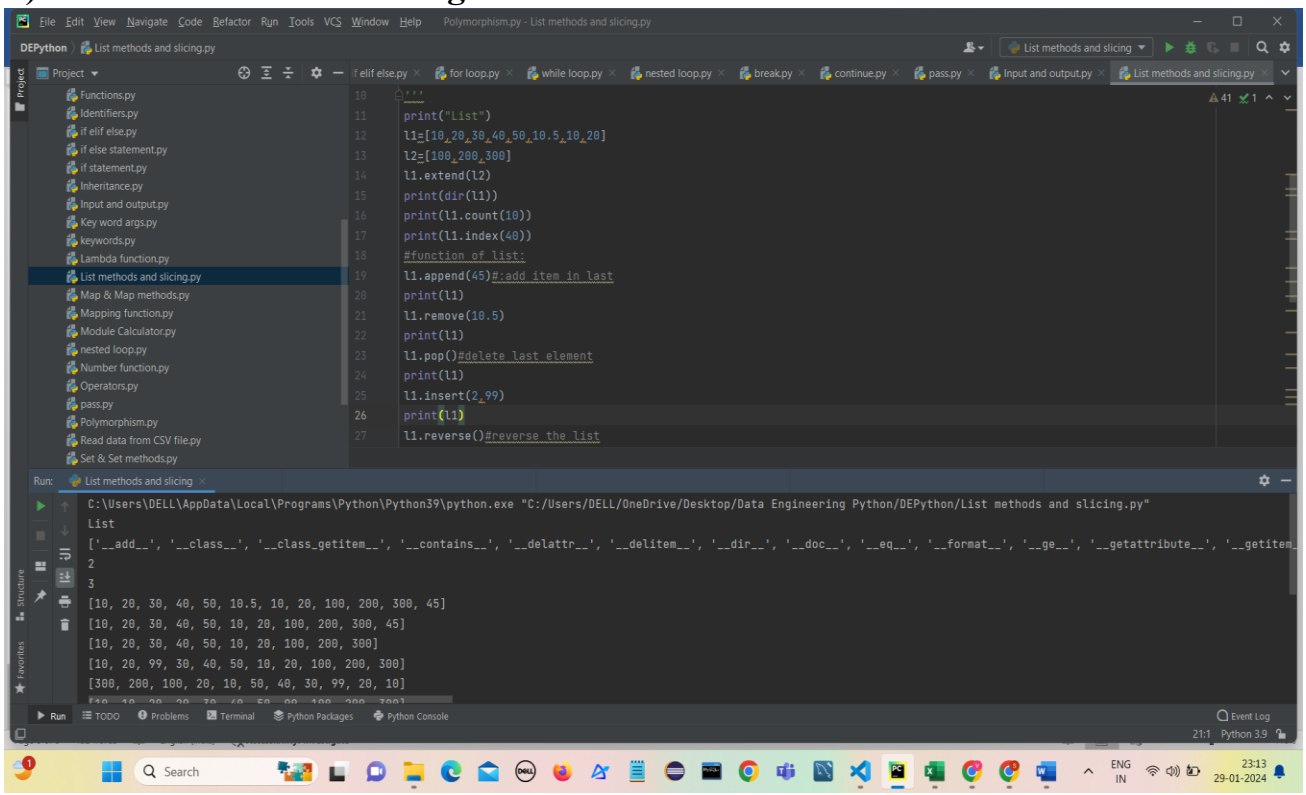
15) Pass

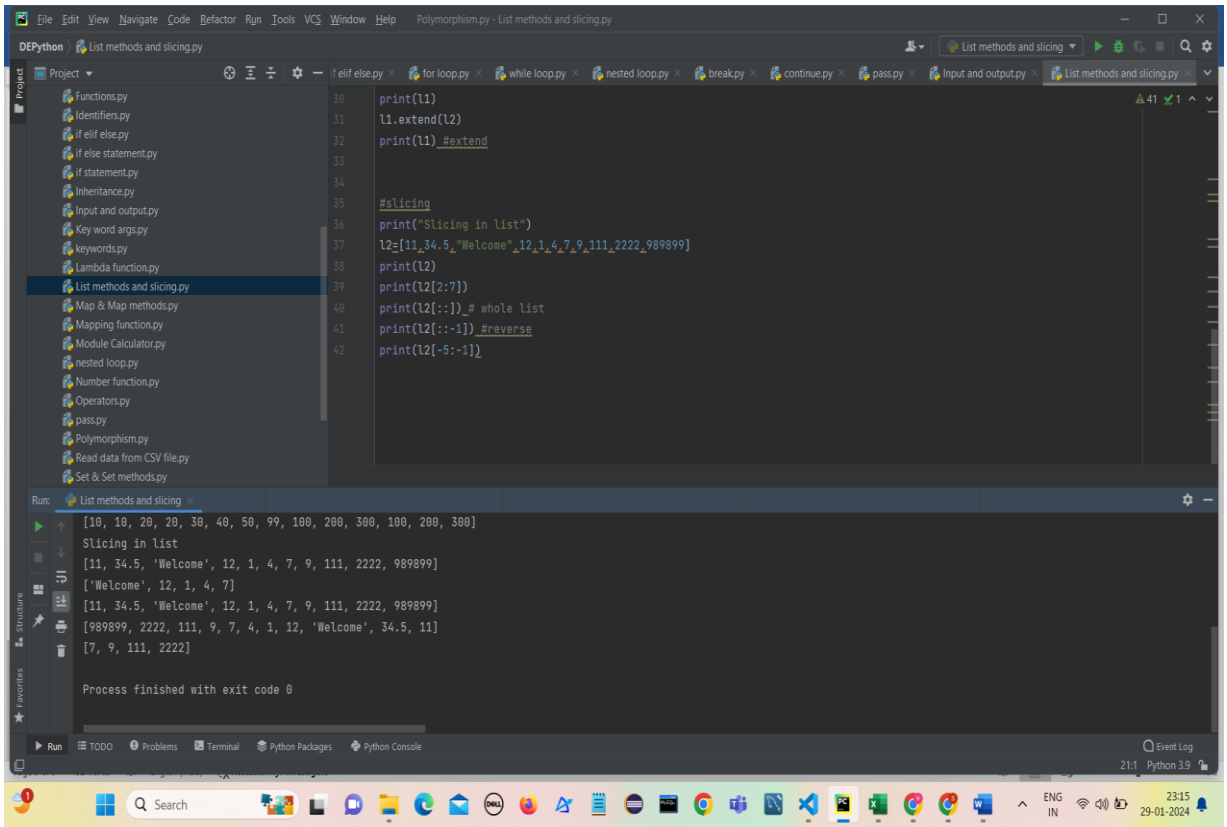


16) Input and Output



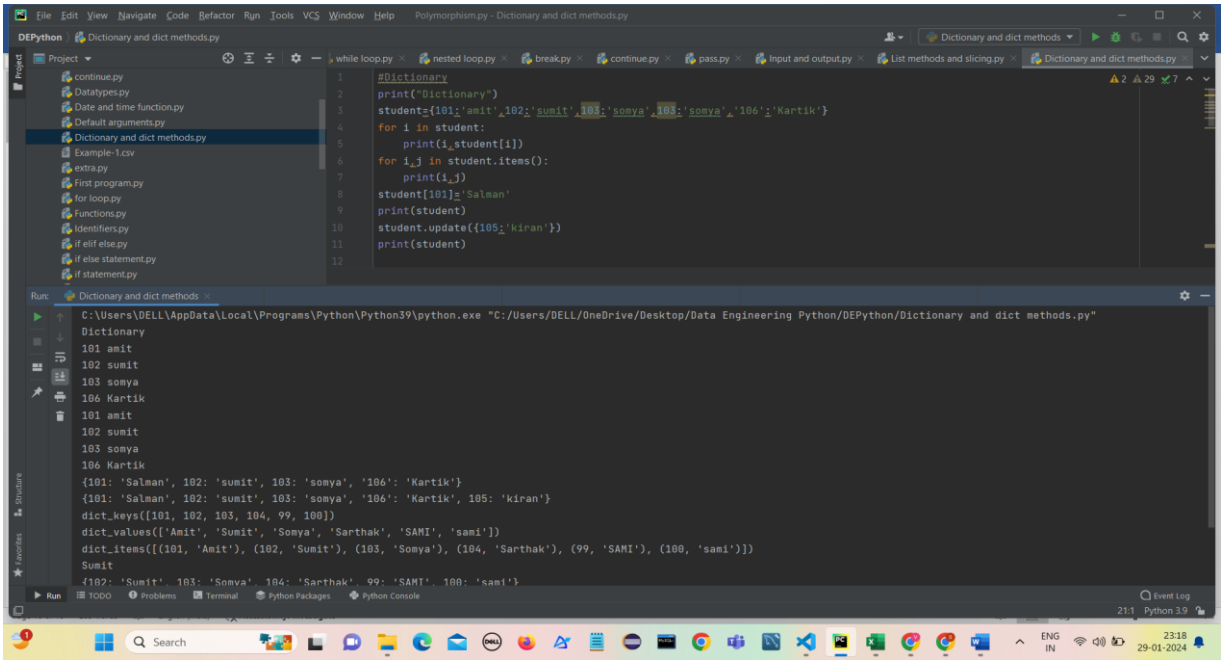
17) List methods and Slicing

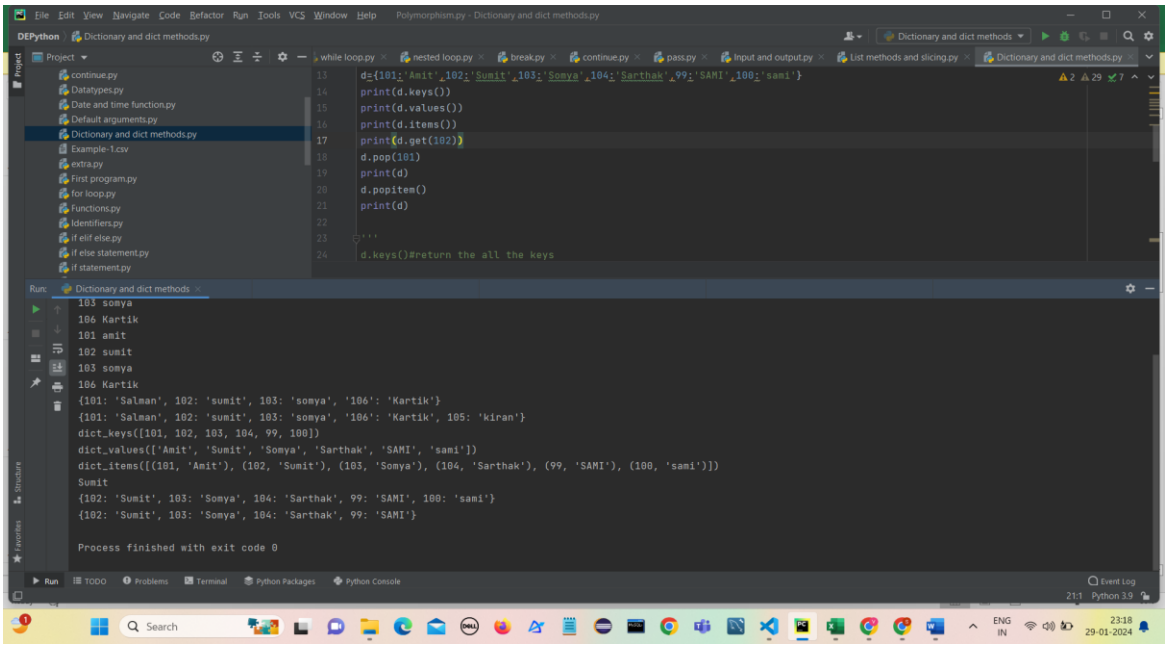




18) Dictionary and Dictionary methods

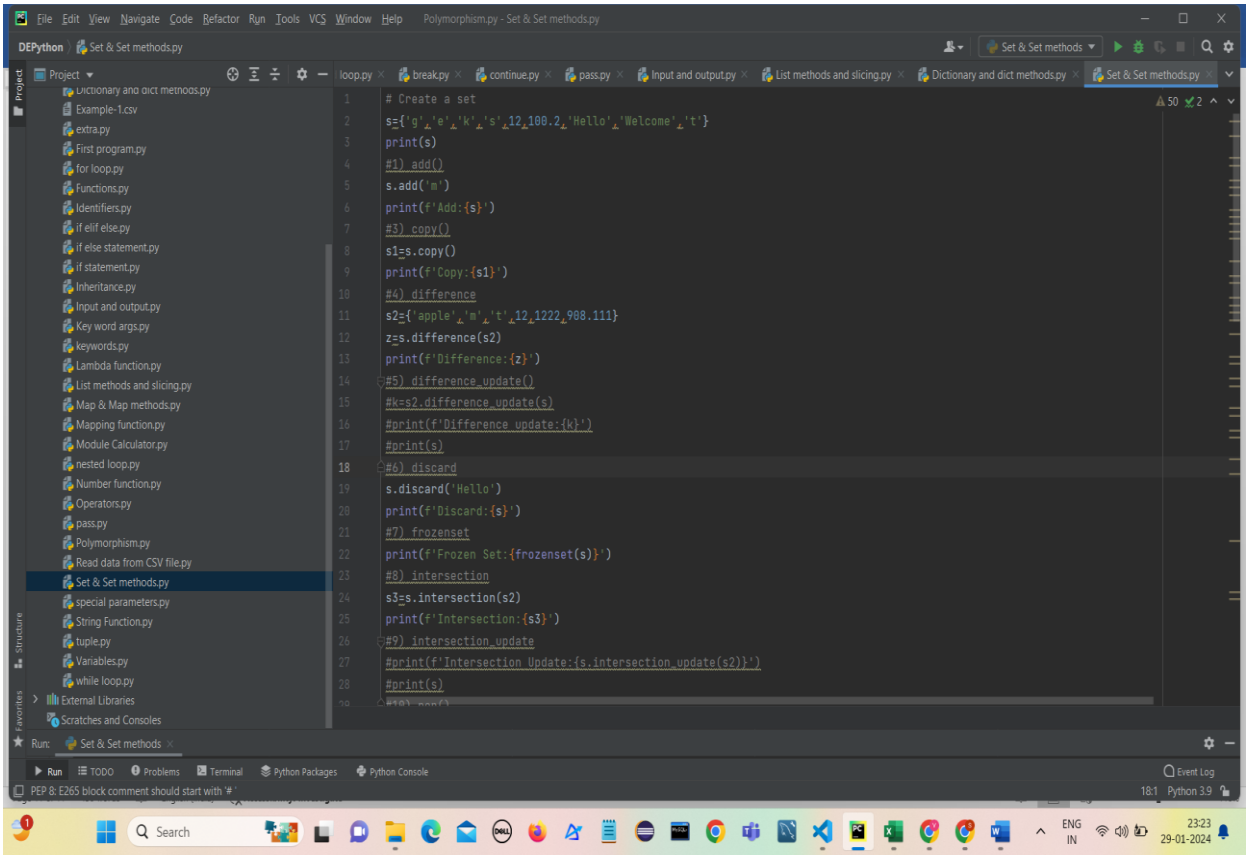
```
'''
d.keys()#return the all the keys
d.values()#return the all values
d.items()#return the key and value in pair
d.clear()#delete all key and values
d.get(101)#return the values
d.update({106:'Likitha'})#add the pair in the dict
d.pop(99)#delete the given key from the dict
d.popitem()#delete the last pair from dict
'''
```

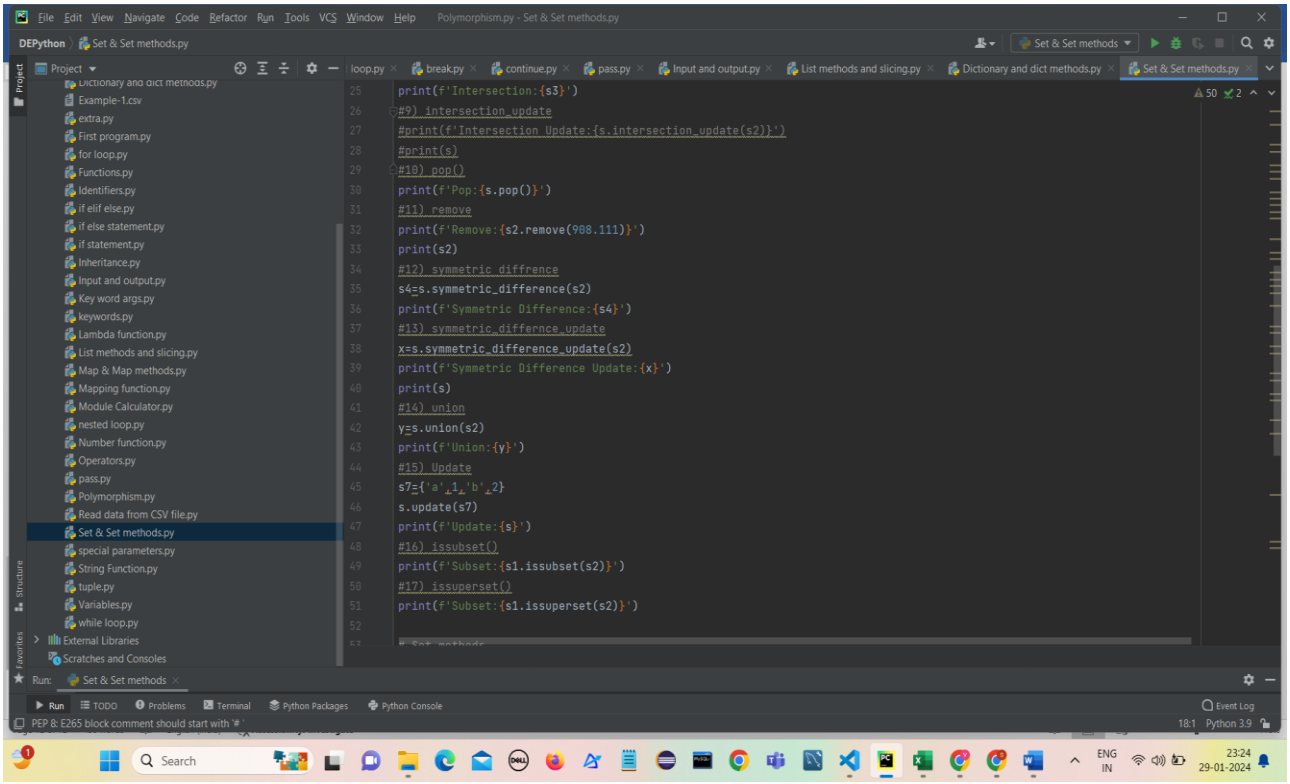




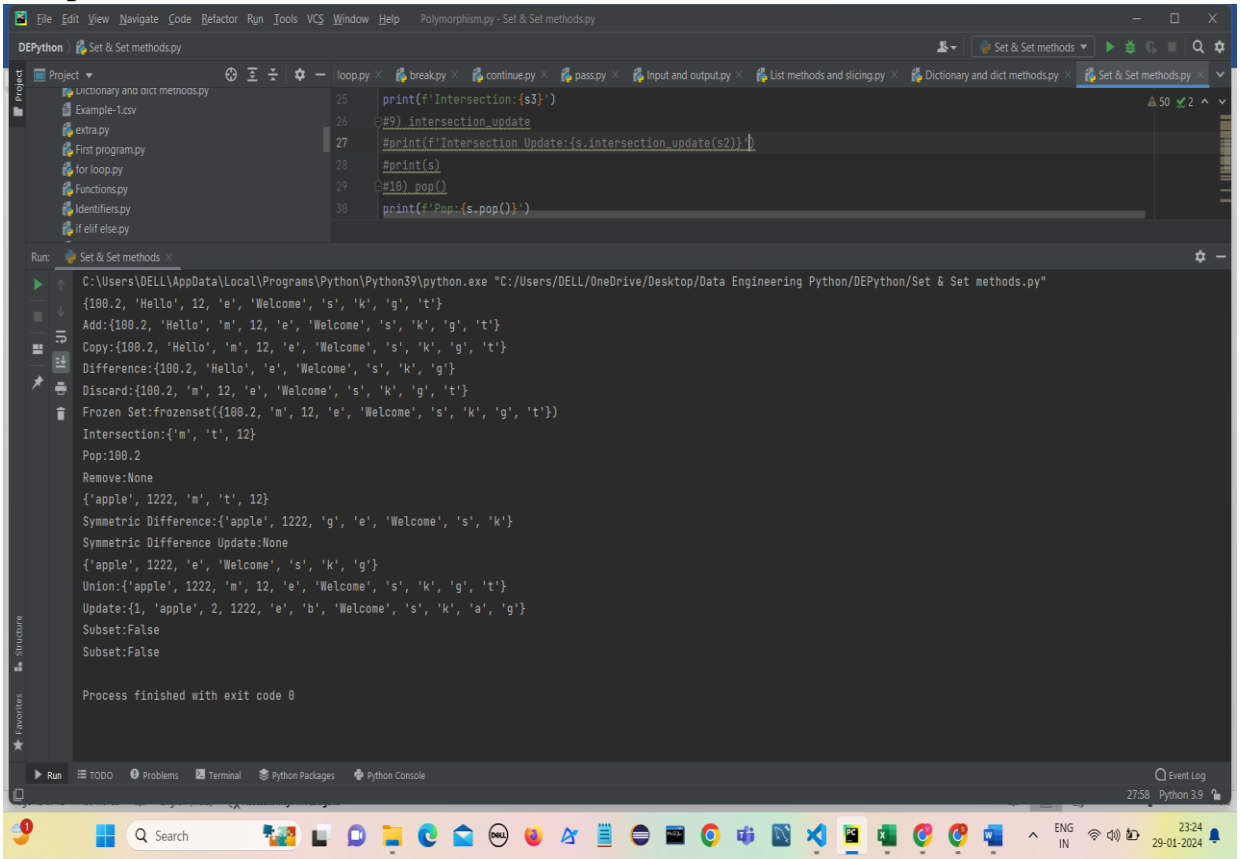
19) Introduction to Set and Set methods

```
# Set methods
'''
1) add()--Adds a given element to a set
2) clear()--Removes all elements from the set
3) copy()--Returns a shallow copy of the set
4) difference()--Returns a set that is the difference between two
sets(returns first set without common elements)
5) difference_update()--Updates the existing caller set with the difference
between two sets(removes items existing in both sets)
6) discard()--Removes the element from the set
7) frozenset()--Return an immutable frozenset object
8) intersection()--Returns a set that has the intersection of all sets
9) intersection_update()--Updates the existing caller set with the
intersection of sets
10) pop()--Returns and removes a random element from the set
11) remove()--Removes the element from the set
12) symmetric_difference()--returns a set that contains all items from both
set, but not the items that are present in both sets.
13) symmetric_difference_update()--updates the original set by removing
items that are present in both sets, and inserting the other items.
14) union()--Returns a set that has the union of all sets
15) update()--updates the current set, by adding items from another set
16) issubset()--Returns True if all elements of a set A are present in
another set B
17) issuperset()--Returns True if all elements of a set A occupies set B
'''
```

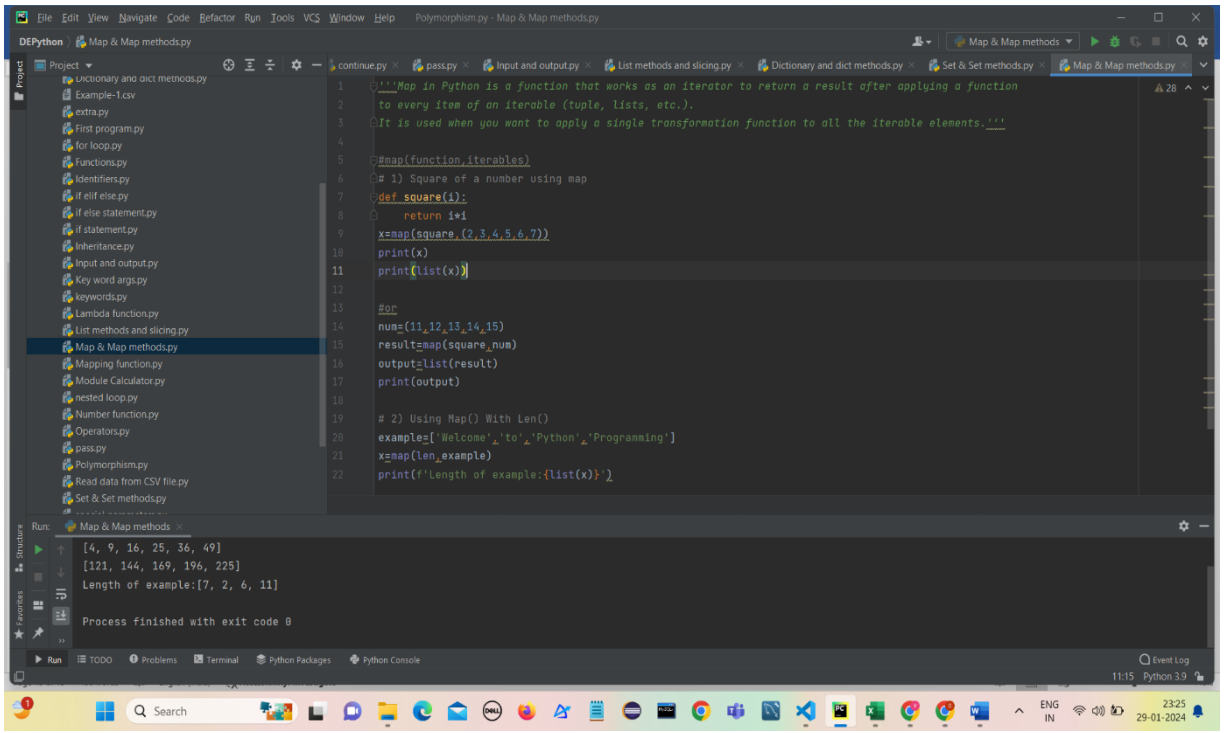



print(f'Intersection:{s3}')
#9) intersection_update
#print(f'Intersection Update:{s.intersection_update(s2)}')
#print(s)
#10) pop()
print(f'Pop:{s.pop()}')
#11) remove
print(f'Remove:{s2.remove(988.111)}')
print(s2)
#12) symmetric_diffrence
s4=s.symmetric_difference(s2)
print(f'Symmetric Difference:{s4}')
#13) symmetric_differnce_update
x=s.symmetric_difference_update(s2)
print(f'Symmetric Difference Update:{x}')
print(s)
#14) union
y=s.union(s2)
print(f'Union:{y}')
#15) Update
s7={'a',1,'b',2}
s.update(s7)
print(f'Update:{s}')
#16) issubset()
print(f'Subset:{s1.issubset(s2)}')
#17) issuperset()
print(f'Subset:{s1.issuperset(s2)}')

Output:


C:\Users\DELL\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/DELL/OneDrive/Desktop/Data Engineering Python/DEPython/Set & Set methods.py"
{100.2, 'Hello', 12, 'e', 'Welcome', 's', 'k', 'g', 't'}
Add:{100.2, 'Hello', 'm', 12, 'e', 'Welcome', 's', 'k', 'g', 't'}
Copy:{100.2, 'Hello', 'm', 12, 'e', 'Welcome', 's', 'k', 'g', 't'}
Difference:{100.2, 'Hello', 'e', 'Welcome', 's', 'k', 'g'}
Discard:{100.2, 'm', 12, 'e', 'Welcome', 's', 'k', 'g', 't'}
Frozen Set:frozenset({100.2, 'm', 12, 'e', 'Welcome', 's', 'k', 'g', 't'})
Intersection:{'m', 't', 12}
Pop:100.2
Remove:None
{'apple', 1222, 'm', 't', 12}
Symmetric Difference:{'apple', 1222, 'g', 'e', 'Welcome', 's', 'k'}
Symmetric Difference Update:None
{'apple', 1222, 'e', 'Welcome', 's', 'k', 'g'}
Union:{'apple', 1222, 'm', 12, 'e', 'Welcome', 's', 'k', 'g', 't'}
Update:{1, 'apple', 2, 1222, 'e', 'b', 'Welcome', 's', 'k', 'a', 'g'}
Subset:False
Subset:False
Process finished with exit code 0

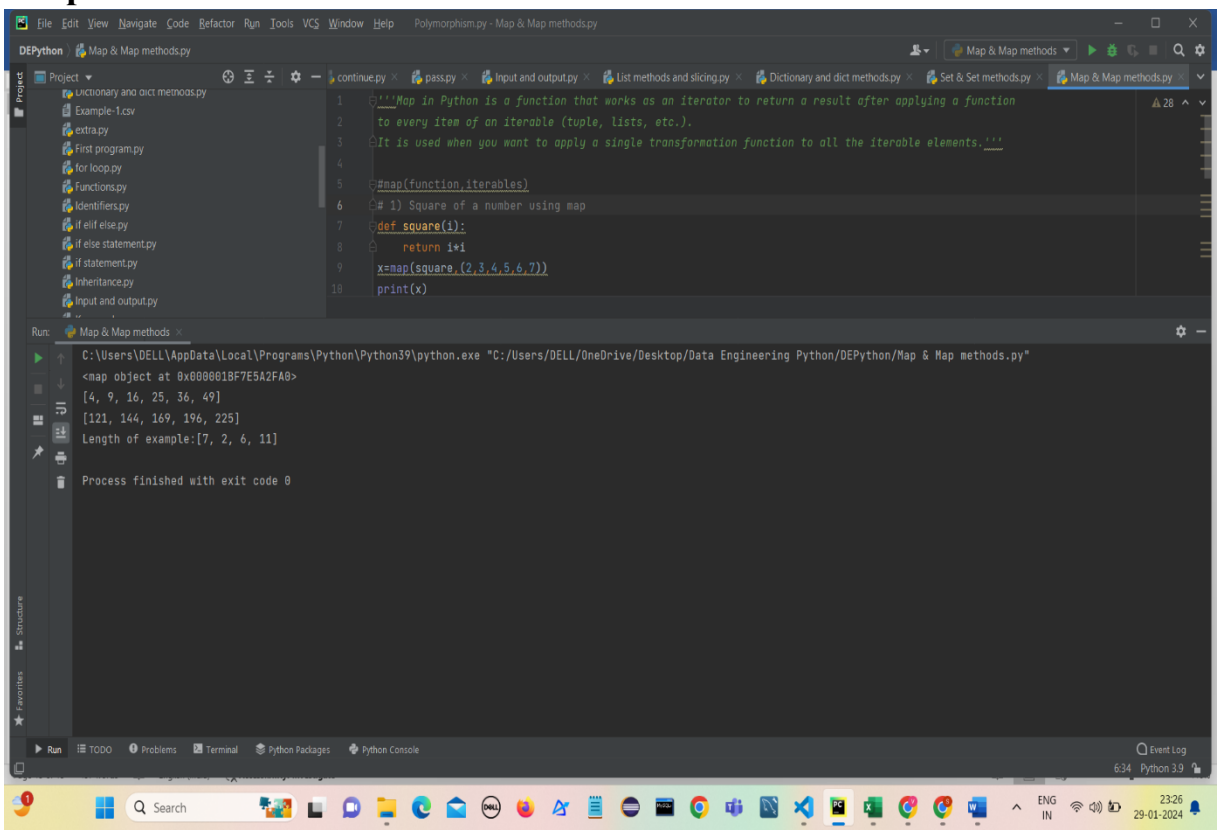
20) Map and Map methods



```
'''Map in Python is a function that works as an iterator to return a result after applying a function
to every item of an iterable (tuple, lists, etc.).
It is used when you want to apply a single transformation function to all the iterable elements.'''

#map(function,iterables)
# 1) Square of a number using map
def square(i):
    return i*i
x=map(square,(2,3,4,5,6,7))
print(x)
print(list(x))

#or
num=(11,12,13,14,15)
result=map(square,num)
output=list(result)
print(output)

# 2) Using Map() With Len()
example=['Welcome','to','Python','Programming']
x=map(len,example)
print(f'Length of example:{list(x)}')
```

```
[4, 9, 16, 25, 36, 49]
[121, 144, 169, 196, 225]
Length of example:[7, 2, 6, 11]

Process finished with exit code 0
```

Output:

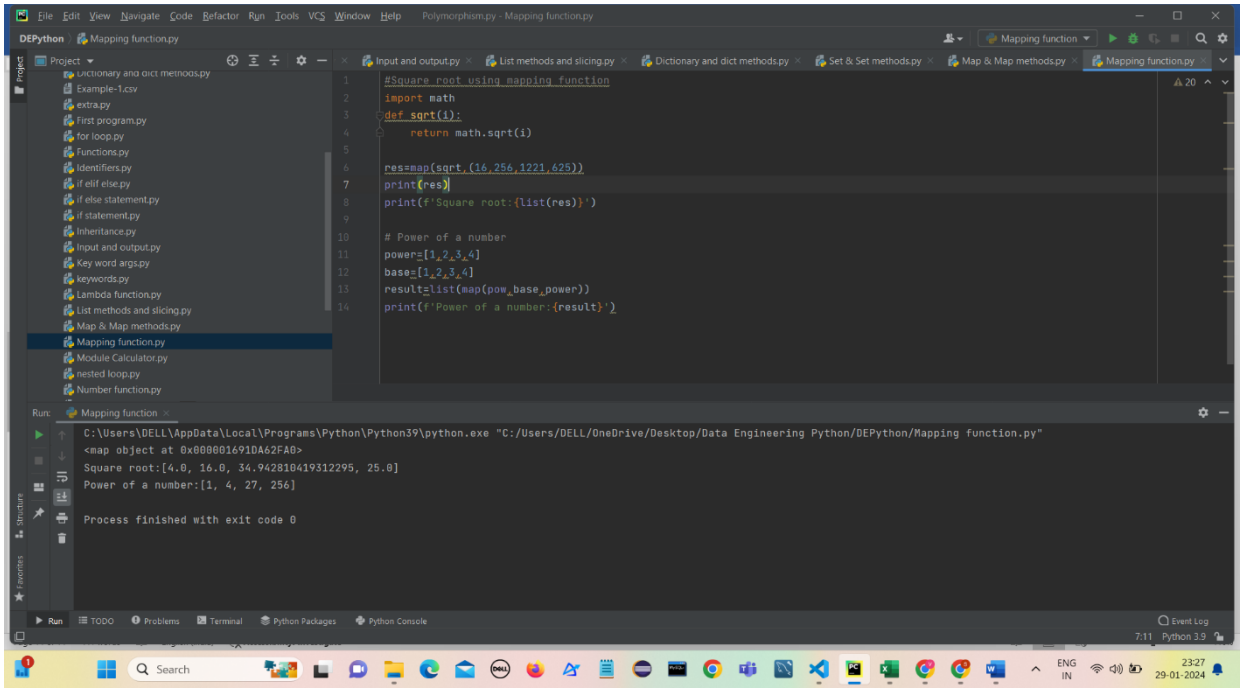


```
'''Map in Python is a function that works as an iterator to return a result after applying a function
to every item of an iterable (tuple, lists, etc.).
It is used when you want to apply a single transformation function to all the iterable elements.'''

#map(function,iterables)
# 1) Square of a number using map
def square(i):
    return i*i
x=map(square,(2,3,4,5,6,7))
print(x)
```

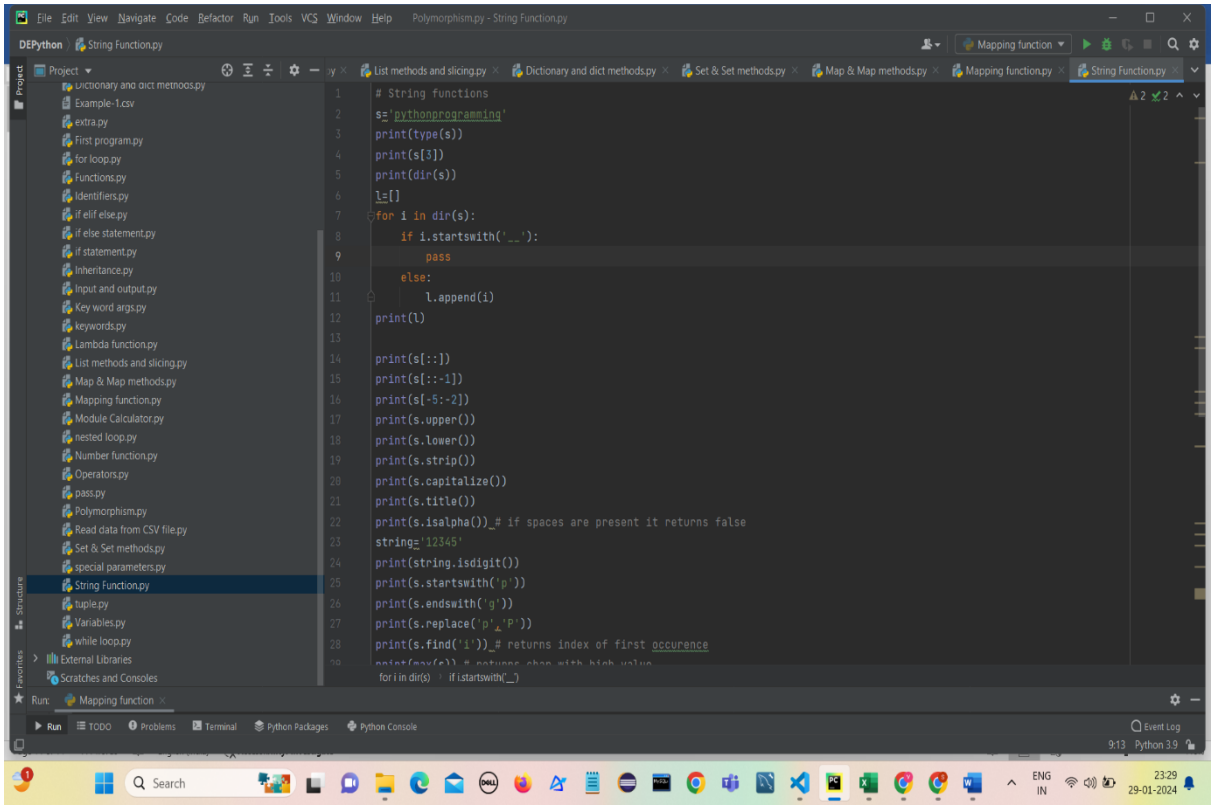
```
C:\Users\DELL\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/DELL/OneDrive/Desktop/Data Engineering Python/DEPython/Map & Map methods.py"
<map object at 0x0000018F7E5A2FA0>
[4, 9, 16, 25, 36, 49]
[121, 144, 169, 196, 225]
Length of example:[7, 2, 6, 11]

Process finished with exit code 0
```

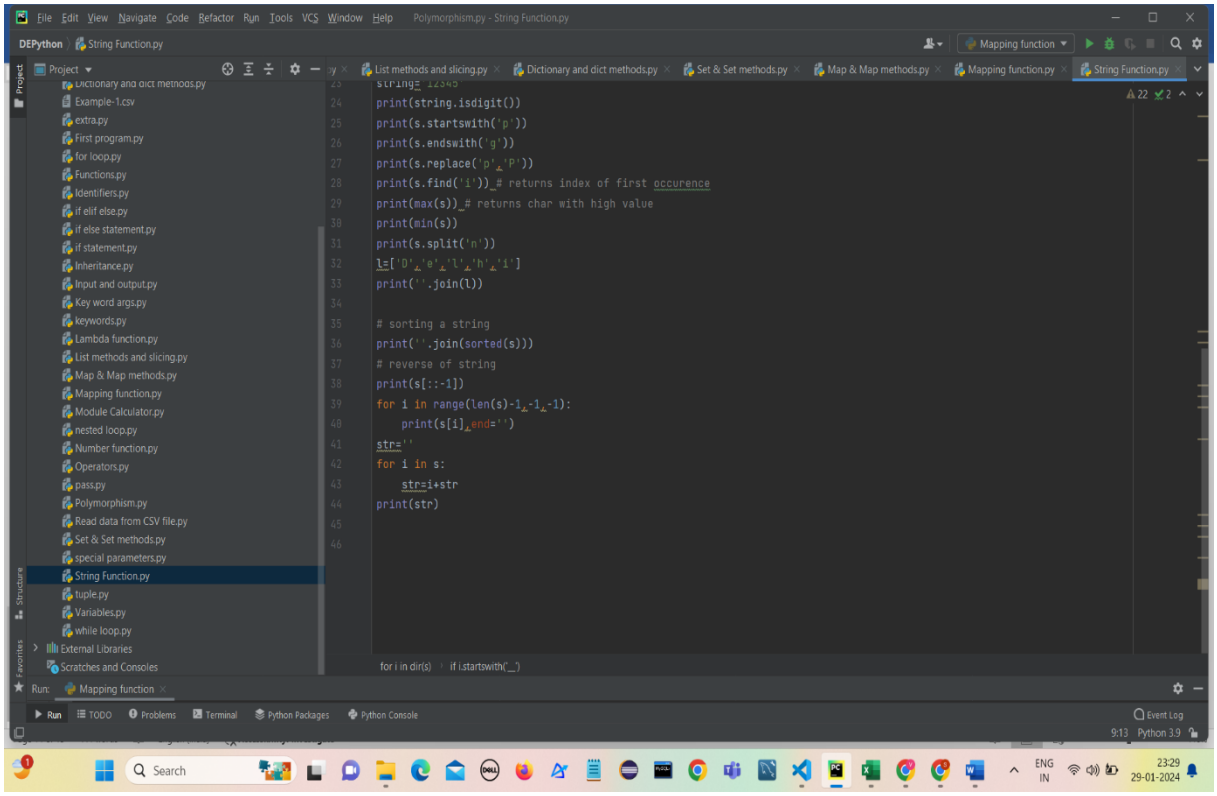
Functions

21) Mapping Function

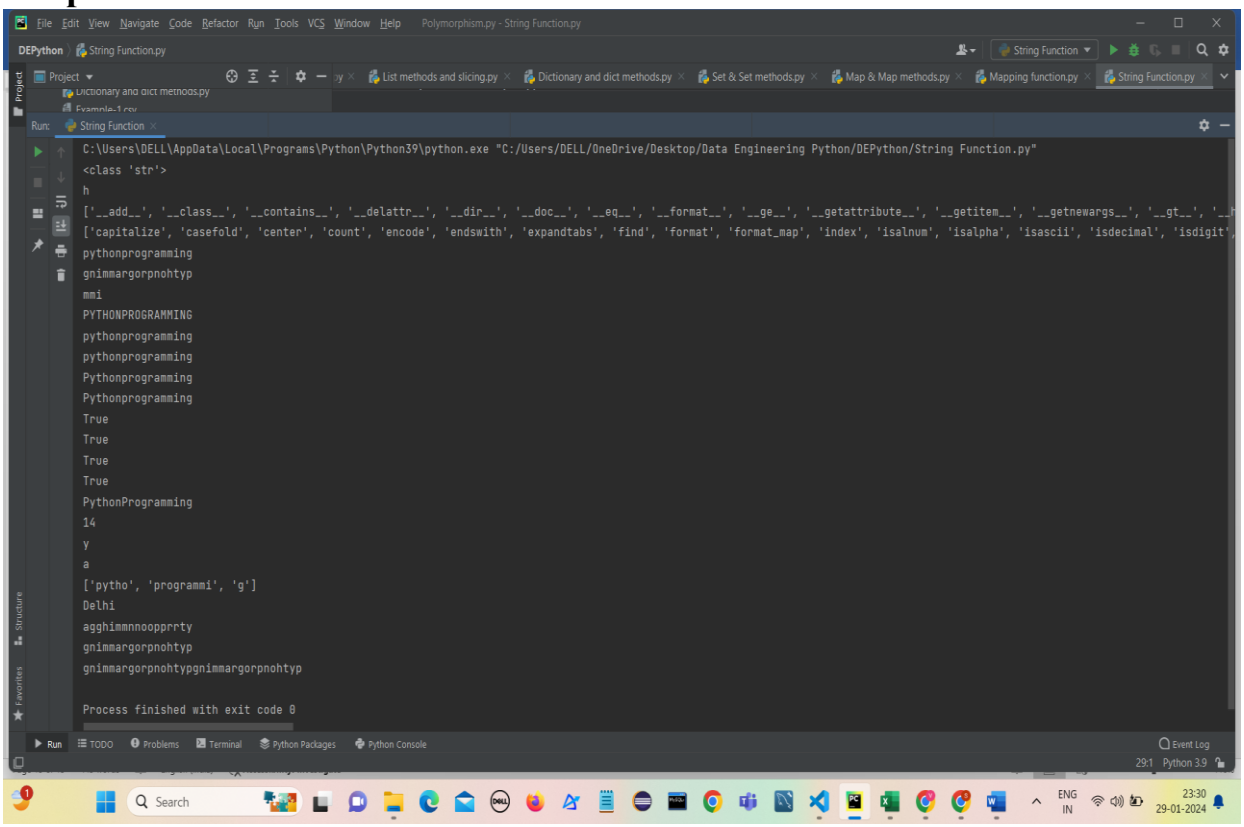


22) String Function

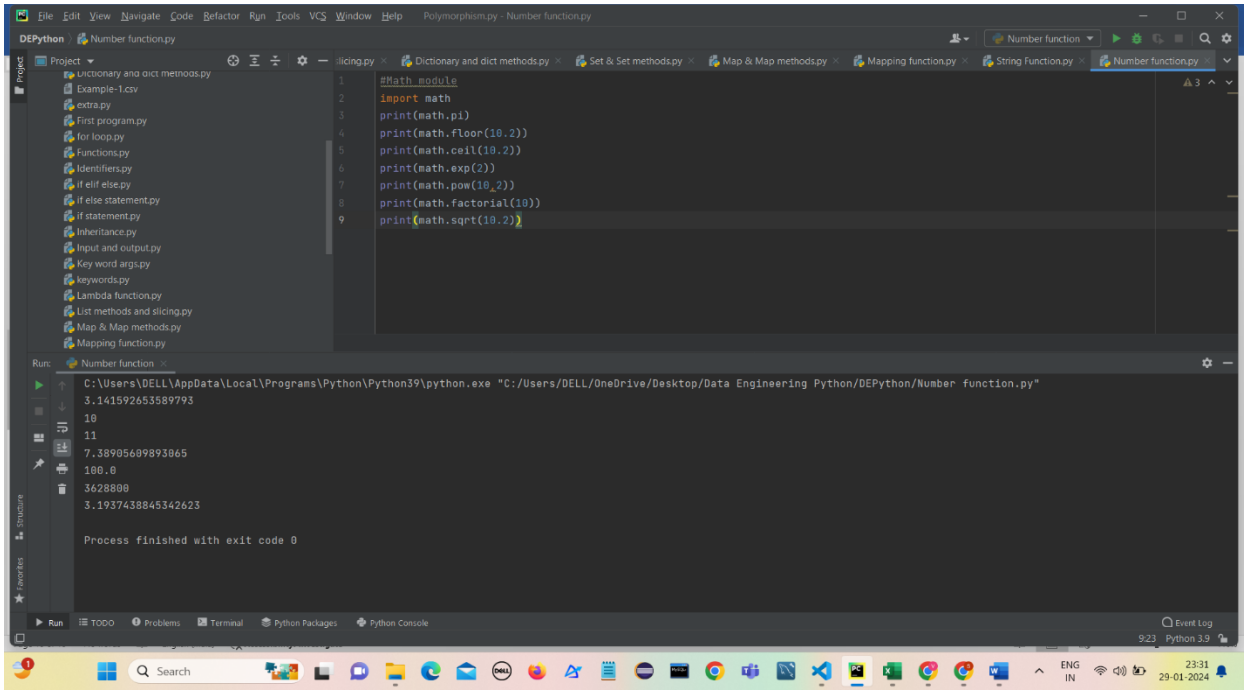



File Edit View Navigate Code Refactor Run Tools VCS Window Help Polymorphism.py - String Function.py
DEPython String Function.py
Mapping function
List methods and slicing.py
Dictionary and dict methods.py
Set & Set methods.py
Map & Map methods.py
Mapping function.py
String Function.py
string= 12345
print(string.isdigit())
print(s.startswith('p'))
print(s.endswith('g'))
print(s.replace('p','P'))
print(s.find('i'))# returns index of first occurence
print(max(s))#returns char with high value
print(min(s))
print(s.split('n'))
l=['D','a','l','h','i']
print(''.join(l))
# sorting a string
print(''.join(sorted(s)))
# reverse of string
print(s[::-1])
for i in range(len(s)-1,-1,-1):
print(s[i],end='')
str=''
for i in s:
str=i+str
print(str)
Run: Mapping function
Run TODO Problems Terminal Python Packages Python Console
Event Log
9:13 Python 3.9
Search
ENG IN
23:29
29-01-2024

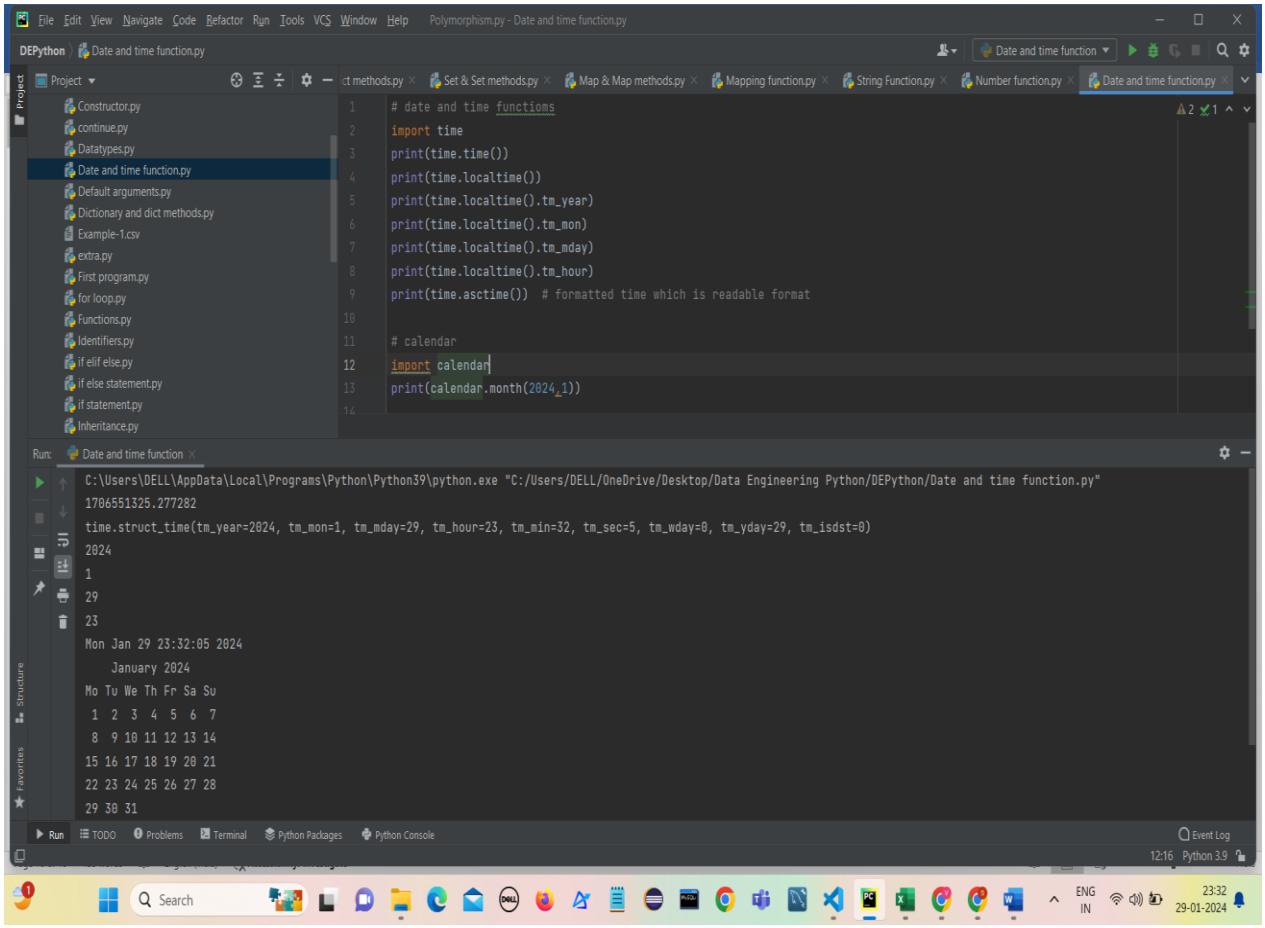
Output:


File Edit View Navigate Code Refactor Run Tools VCS Window Help Polymorphism.py - String Function.py
DEPython String Function.py
String Function
Run: String Function
C:\Users\DELL\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/DELL/OneDrive/Desktop/Data Engineering Python/DEPython/String Function.py"
<class 'str'>
h
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__',
['capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit',
pythonprogramming
gnimmargorpnohtyp
mmi
PYTHONPROGRAMMING
pythonprogramming
pythonprogramming
Pythonprogramming
Pythonprogramming
True
True
True
True
PythonProgramming
14
y
a
['pytho', 'programmi', 'g']
Delhi
agghimmnnooppprrty
gnimmargorpnohtyp
gnimmargorpnohtypgnimmargorpnohtyp
Process finished with exit code 0
Run TODO Problems Terminal Python Packages Python Console
Event Log
29:1 Python 3.9
Search
ENG IN
23:30
29-01-2024

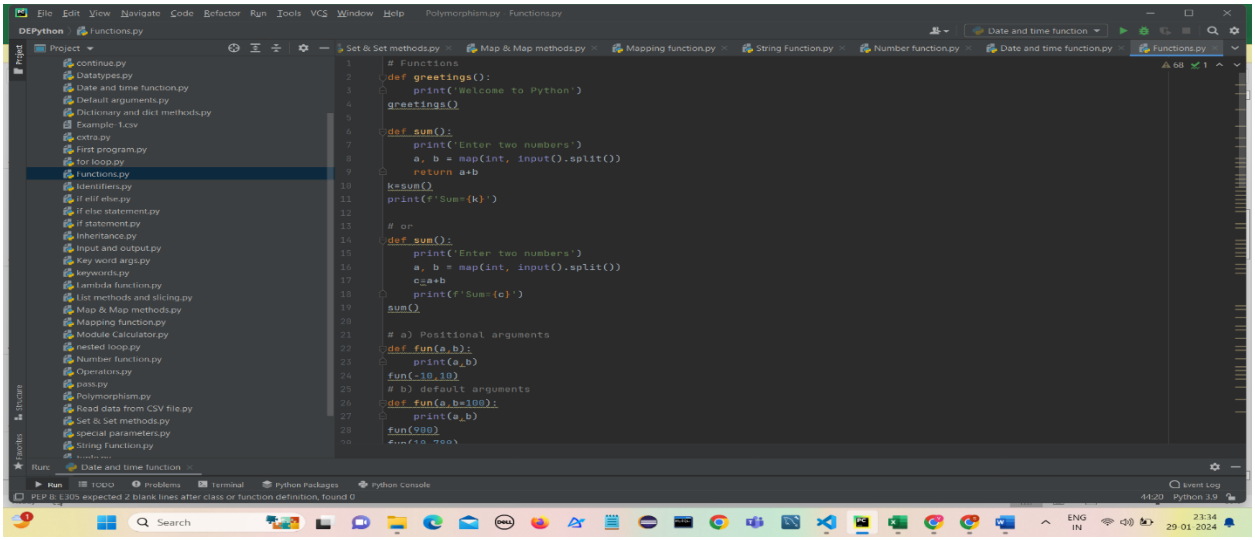
23) Number Function



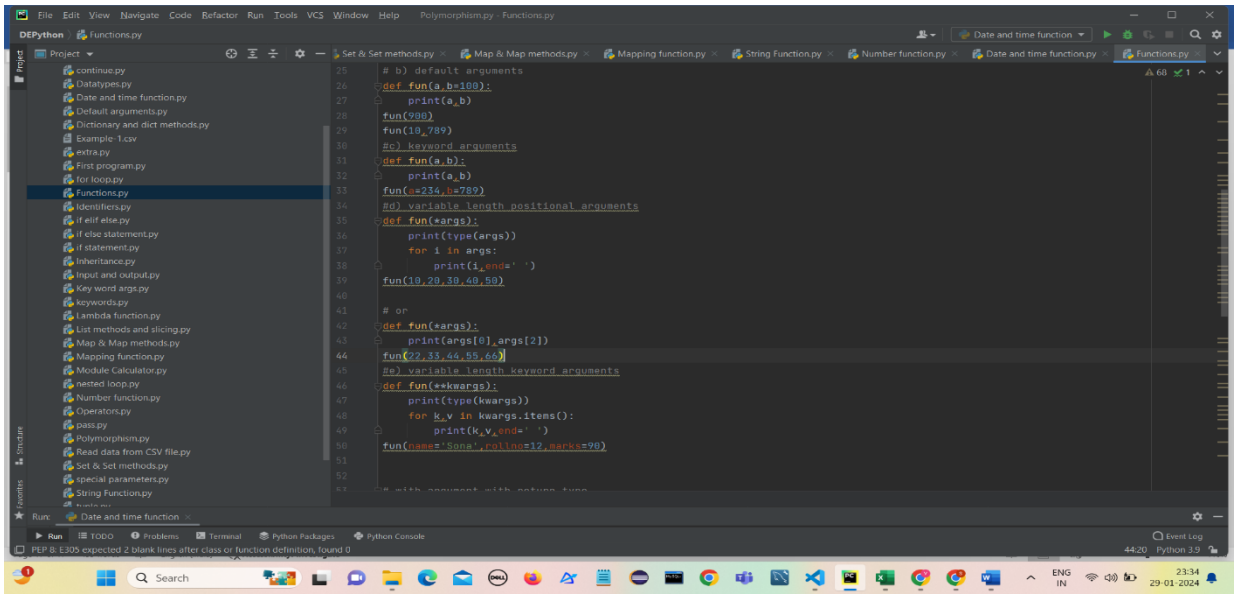
24) Date and Time Function



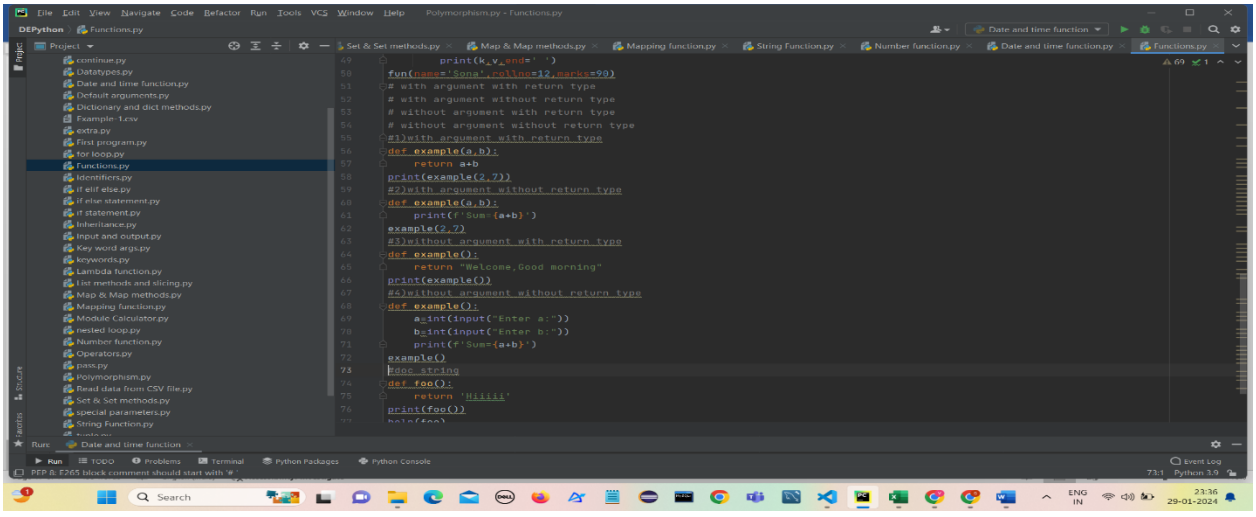
25) Functions



```
# Functions
def greetings():
    print('Welcome to Python')
greetings()

def sum():
    print('Enter two numbers')
    a, b = map(int, input().split())
    return a+b
k=sum()
print(f'Sum={k}')

# or
def sum():
    print('Enter two numbers')
    a, b = map(int, input().split())
    c=a+b
    print(f'Sum={c}')
sum()

# a) Positional arguments
def fun(a,b):
    print(a,b)
fun(-10,10)
# b) default arguments
def fun(a,b=100):
    print(a,b)
fun(900)
```

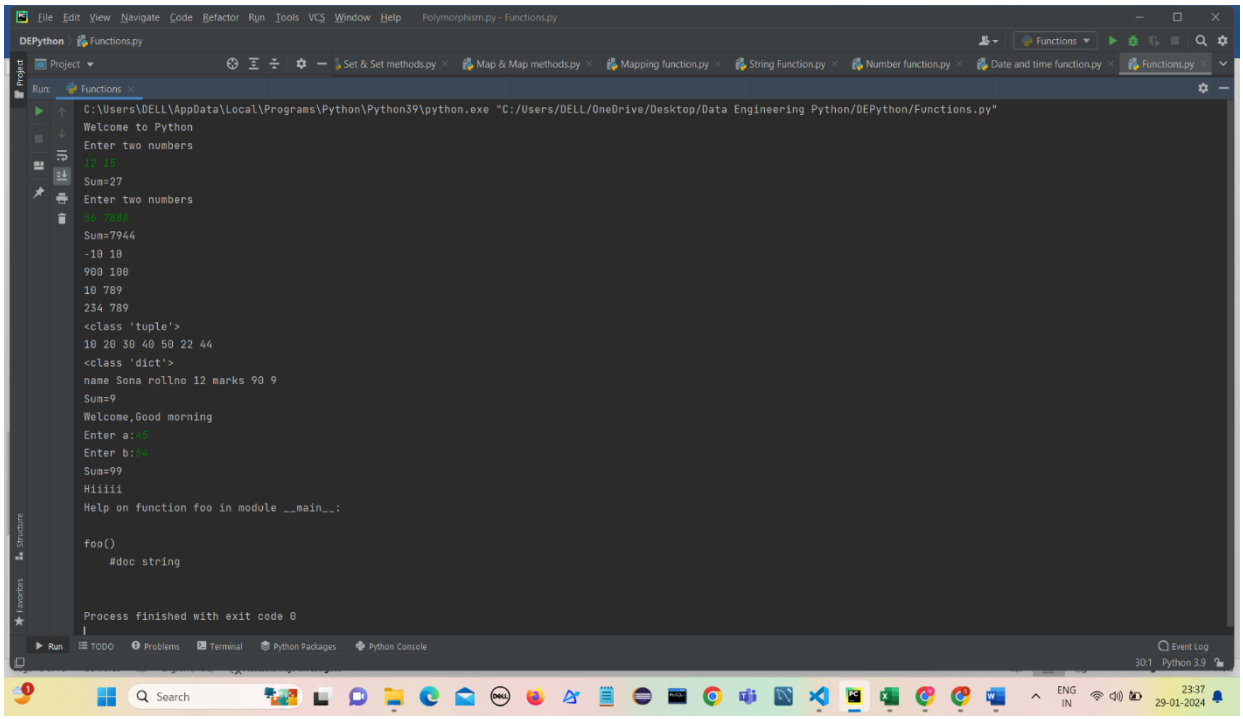


```
# b) default arguments
def fun(a,b=100):
    print(a,b)
fun(900)
fun(10,789)
#c) keyword arguments
def fun(a,b):
    print(a,b)
fun(a=234,b=789)
#d) variable length positional arguments
def fun(*args):
    print(type(args))
    for i in args:
        print(i,end=' ')
fun(10,20,30,40,50)

# or
def fun(*args):
    print(args[0],args[2])
fun(22,33,44,55,66)
#e) variable length keyword arguments
def fun(**kwargs):
    print(type(kwargs))
    for k,v in kwargs.items():
        print(k,v,end=' ')
fun(name='Sona',rollno=12,marks=90)
```

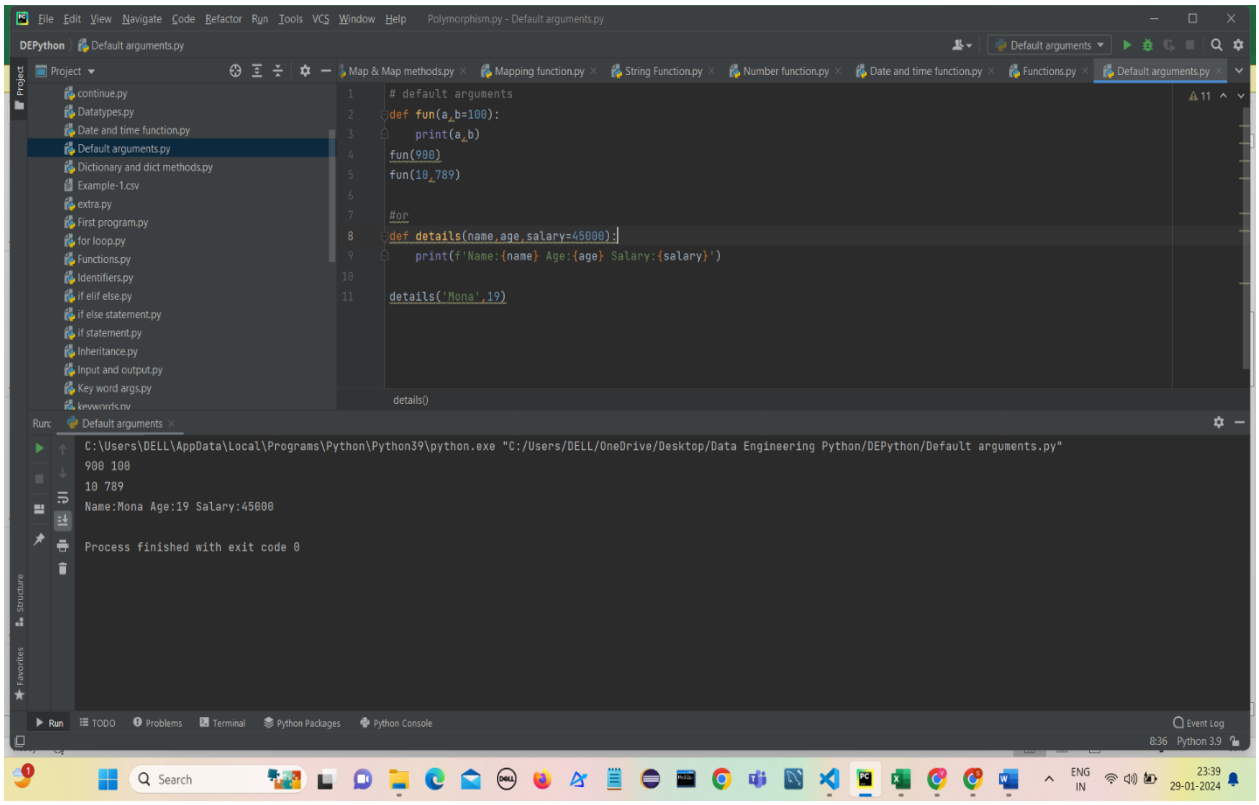


```
        print(k,v,end=' ')
fun(name='Sona',rollno=12,marks=90)
# with argument with return type
# with argument without return type
# without argument with return type
# without argument without return type
#1)with argument with return type
def example(a,b):
    return a+b
print(example(2,7))
#2)with argument without return type
def example(a,b):
    print(f'Sum={a+b}')
example(2,7)
#3)without argument with return type
def example():
    return "Welcome,Good morning"
print(example())
#4)without argument without return type
def example():
    a=int(input("Enter a:"))
    b=int(input("Enter b:"))
    print(f'Sum={a+b}')
example()
#doc string
def foo():
    return 'Hiiii'
print(foo())
```

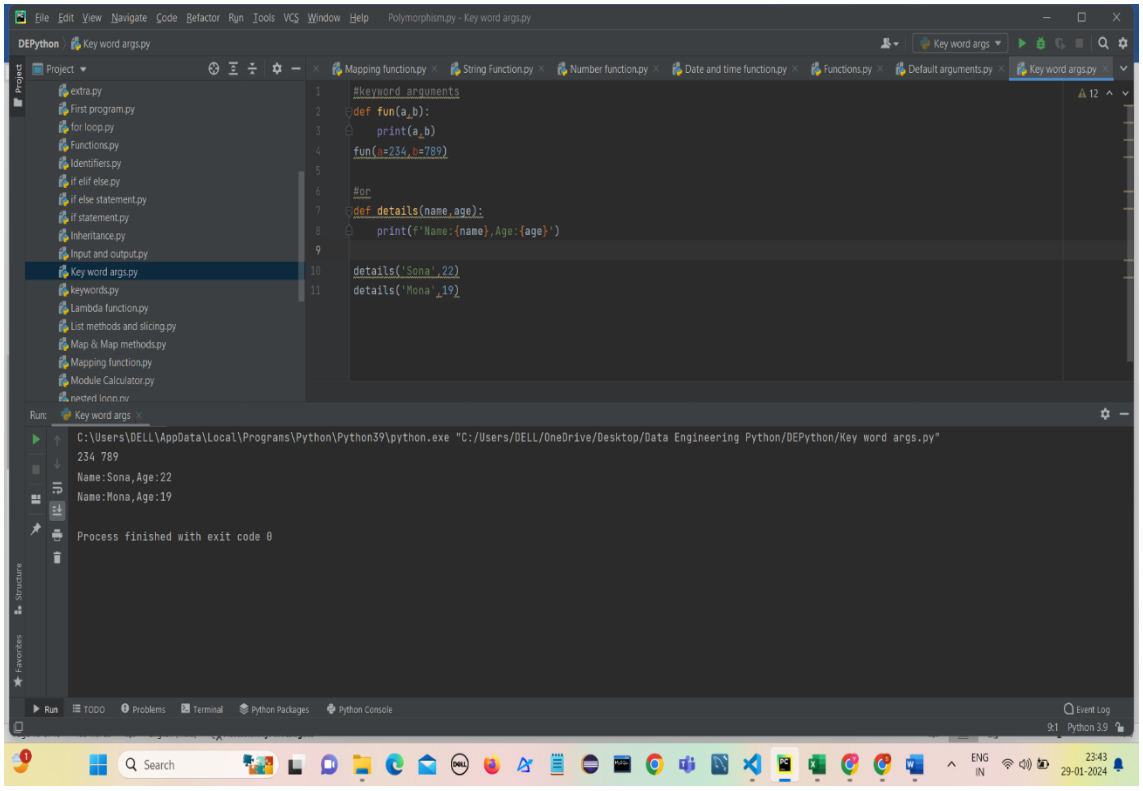
Output:



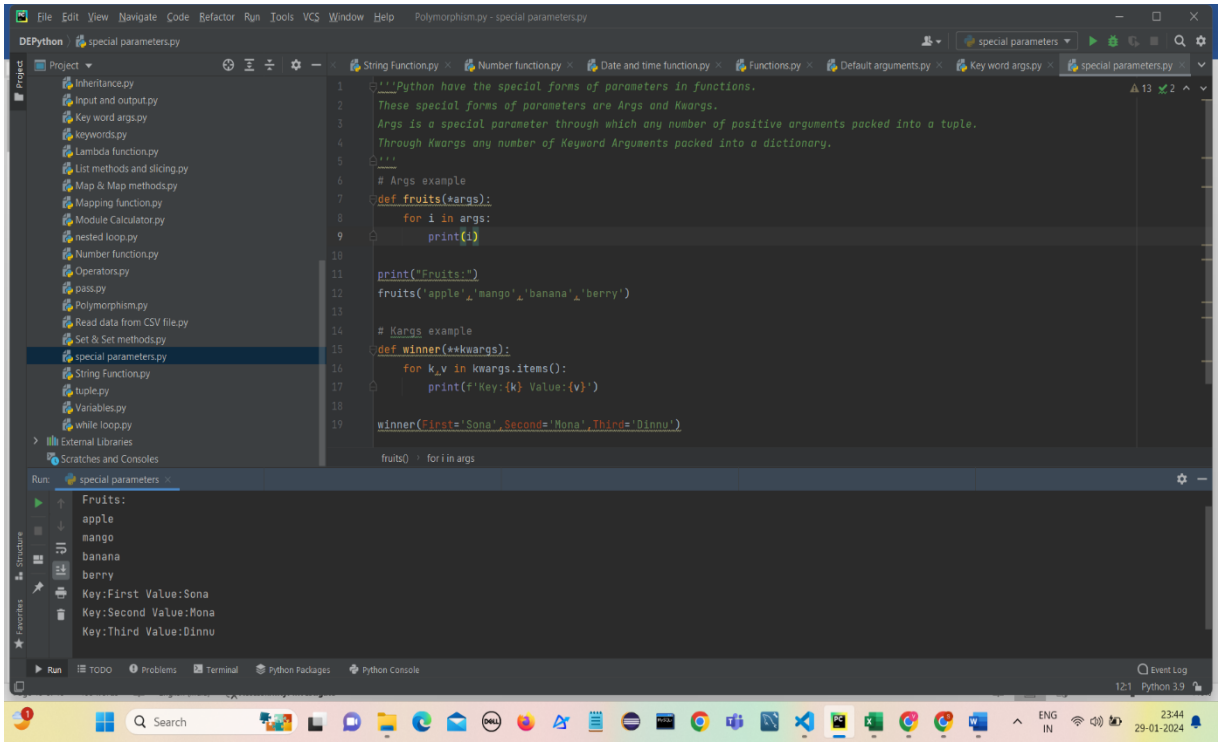
26) Default Argument Values



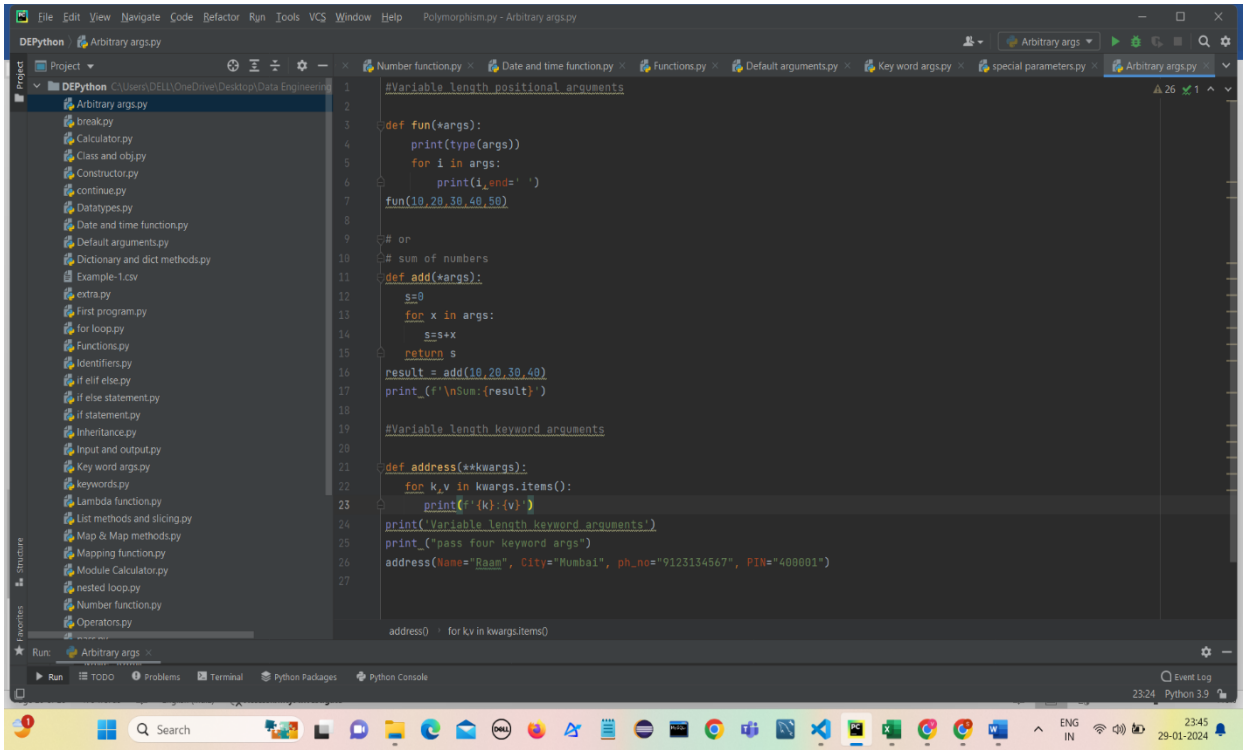
27) Keyword arguments

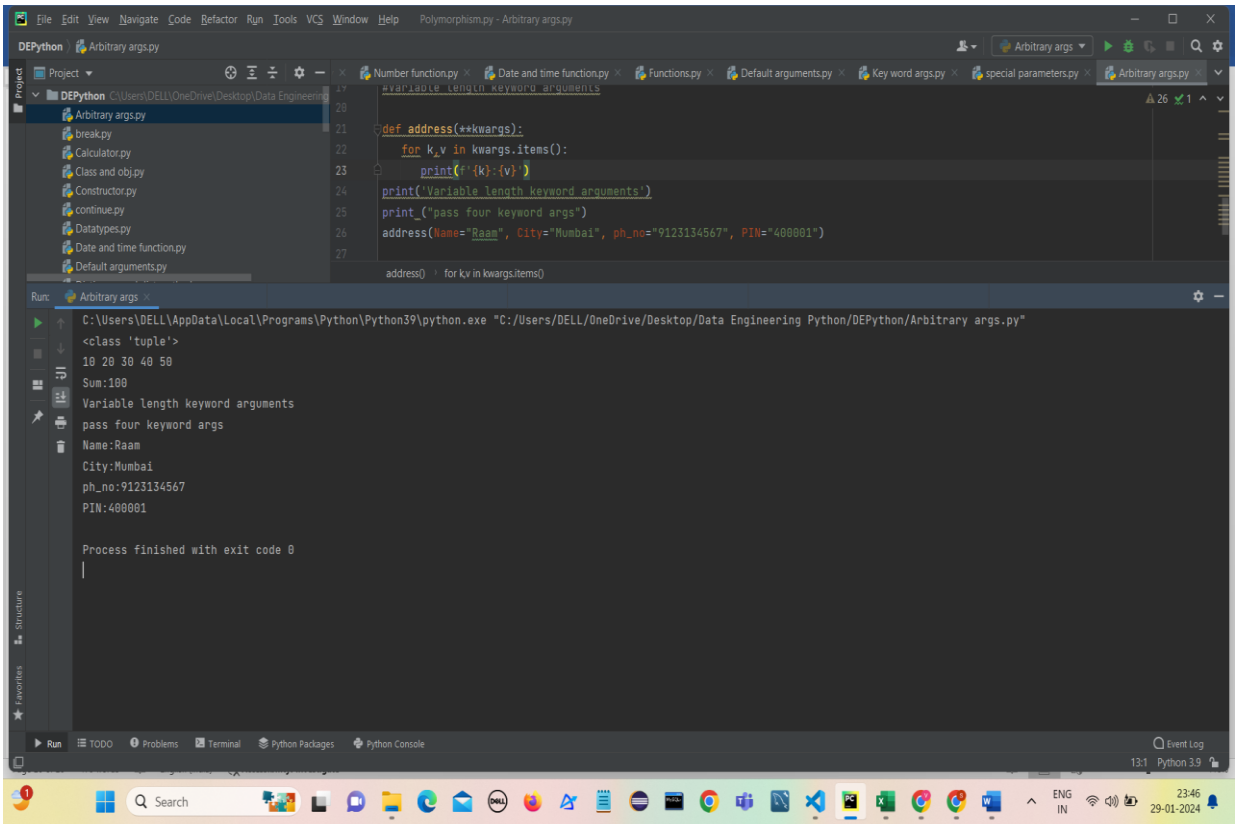


28) Special Parameters



29) Arbitrary argument lists





30) Lambda Expressions