Name: Rohan Vinayak Chaudhari

Batch: Data Engineering

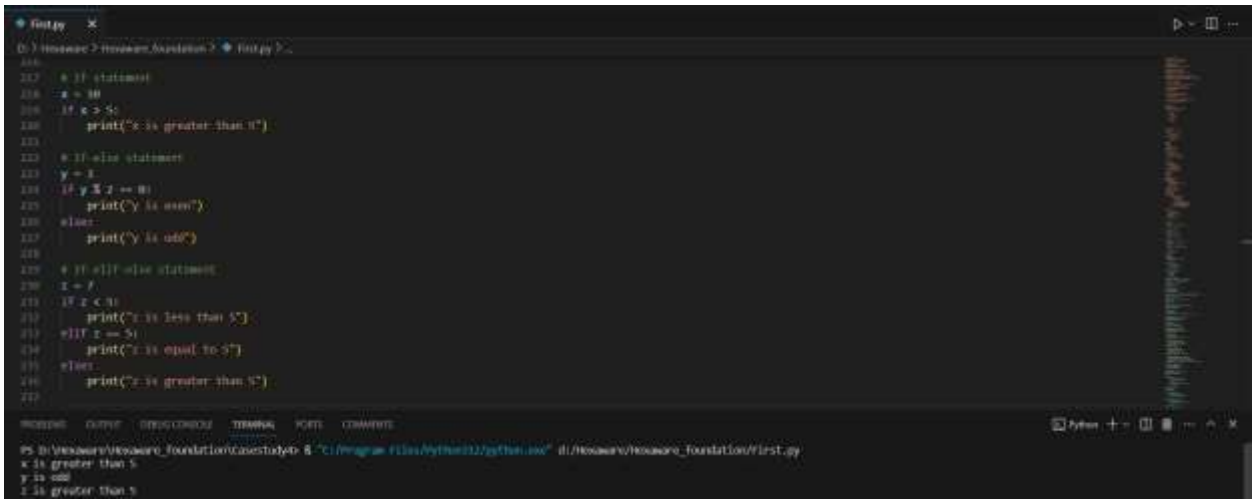
Date:29/01/2024

Topic: Python(control structures,list,sets,tuple,dict,oops,reading csv)

Solution:

1.Python:

1)If,ifelse,nested if elif:



2)for,while,nestedloop,break,continue,pass

First.py

D: > Hexaware > Hexaware_foundation > First.py > ...

```python
# For Loop
for i in range(5):
    print(i)

# While Loop
count = 0
while count < 3:
    print("Count:", count)
    count += 1

# Nested Loop
for i in range(3):
    for j in range(2):
        print(i, j)

# Break, Continue & Pass
for num in range(5):
    if num == 3:
        break
    print(num)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
0
1
2
3
4
Count: 0
Count: 1
Count: 2
0 0
0 1
1 0
1 1
2 0
2 1
0
1
2
```

3)input,list,list methods,set

First.py

D: > Hexaware > Hexaware_foundation > First.py > ...

```python
# Input and Output
user_input = input("Enter something: ")
print("You entered:", user_input)

# Introduction to Lists
my_list = [1, 2, 3, 4]
print(my_list)

# List Methods and Slicing
my_list.append(5)
print(my_list[1:3])

# Introduction to Dictionaries & Dictionary Methods
my_dict = {'key1': 'value1', 'key2': 'value2'}
print(my_dict['key1'])

# Introduction to Set & Set Methods
my_set = {1, 2, 3, 3}
print(my_set)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
Enter something: 1
You entered: 1
[1, 2, 3, 4]
[2, 3]
value1
{1, 2, 3}
```

4)map,map methods,lambda function,date time.string function.mapping function,number function

First.py

D: > Hexaware > Hexaware_foundation > First.py > ...

```python
# Introduction to Map & Map Methods
my_map = map(lambda x: x * 2, [1, 2, 3])
print(list(my_map))

# Mapping function
Codeium: Refactor | Explain | Generate Docstring | X
def square(x):
    return x ** 2
squared_values = list(map(square, [1, 2, 3]))

# String Function
string_example = "Hello, World!"
print(len(string_example))

# Number Function
num_example = 3.14
print(round(num_example))

# Date and Time Function
from datetime import datetime
current_time = datetime.now()
print(current_time)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
[2, 4, 6]
13
3
2024-01-29 21:25:51.500858
```

5)functions,different types of functions:

First.py

D: > Hexaware > Hexaware_foundation > First.py > ...

```python
# Python Functions
def greet(name):
    return "Hello, " + name + "!"
print(greet("rohan"))

# Default Argument Values
def greet_default(name="Guest"):
    return "Hello, " + name + "!"
print(greet_default())

# Keyword Arguments
def print_info(name, age):
    print("Name:", name)
    print("Age:", age)
print_info(name="rohit", age=25)

# Special parameters
def special_params(*args, **kwargs):
    print("Positional arguments:", args)
    print("Keyword arguments:", kwargs)
special_params(1, 2, 3, key1="value1", key2="value2")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
Hello, Guest!
Name: rohit
Age: 25
Positional arguments: (1, 2, 3)
Keyword arguments: {'key1': 'value1', 'key2': 'value2'}
```

6)arbitrary argument, oops, class, obj, access specifiers

First.py

D: > Hexaware > Hexaware_foundation > First.py > ...

```python
# Arbitrary Argument Lists
Codeium: Refactor | Explain | Generate Docstring | X
def sum_all(*numbers):
    return sum(numbers)
result = sum_all(1, 2, 3, 4)


# OOPS
Codeium: Explain
class MyClass:
    Codeium: Refactor | Explain | Generate Docstring | X
    def __init__(self, value):
        self.value = value

    Codeium: Refactor | Explain | Generate Docstring | X
    def display(self):
        print("Value:", self.value)

# Class and Object
obj = MyClass(10)
obj.display()

# Access Specifiers
Codeium: Explain
class MyClassPrivate:
    Codeium: Refactor | Explain | Generate Docstring | X
    def __init__(self, value):
        self.__value = value
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
Positional arguments: (1, 2, 3)
Keyword arguments: {'key1': 'value1', 'key2': 'value2'}
Value: 10
```

7)constructor,polymorphism,inheritance

First.py

D: > Hexaware > Hexaware_foundation > First.py > ...

```python
# Constructor
Codeium: Explain
class MyClassConstructor:
    Codeium: Refactor | Explain | Generate Docstring |
    def __init__(self):
        print("Constructor called")

# Inheritance
Codeium: Explain
class ChildClass(MyClass):
    Codeium: Refactor | Explain | Generate Docstring |
    def __init__(self, value, additional_value):
        super().__init__(value)
        self.additional_value = additional_value

# Polymorphism
Codeium: Explain
class sample:
    Codeium: Refactor | Explain | Generate Docstring |
    def __init__(self,*args):
        if len(args) >1:
            self.ans=0
            for i in args:
                self.ans+=i
        elif isinstance(args[0],int):
            self.ans=args[0]*args[0]
        elif isinstance(args[0],str):
            self.ans=len(args[0])

```

8)overriding,file handling,exception handling

First.py

D: > Hexaware > Hexaware_foundation > First.py > ...

```python

# Method Overriding
class parent():
    def greet(Self):
        print('hello a')
class child(parent):
    def greet(Self):
        print('hello b')
p=parent()
c=child()
p.greet()
c.greet()

# File handling
with open("example.txt", "w") as file:
    file.write("Hello, File!")

# Exception Handling
try:
    result = 10 / 0
except ZeroDivisionError as e:
    print("Error:", e)

import math

# Using math module functions
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
Value: 10
hello a
hello b
Error: division by zero
```

9)modules, math module

```
First.py
D: > Hexaware > Hexaware_foundation > First.py > ...

import math

# Using math module functions
print("Square root of 16:", math.sqrt(16))
print("Ceiling of 3.14:", math.ceil(3.14))
print("Factorial of 5:", math.factorial(5))
# Power function
print("2 to the power of 3:", math.pow(2, 3))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
hello b
Error: division by zero
Square root of 16: 4.0
Ceiling of 3.14: 4
Factorial of 5: 120
2 to the power of 3: 8.0
PS D:\Hexaware\Hexaware_foundation\Casestudy4>
```

10)reading CSV FILE:

```
import pandas as pd
```

```
C:\Users\ishad\AppData\Local\Temp\ipykernel_15472\3809738814.py:1: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466
```

```
import pandas as pd
```

```
df=pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv')
```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S