Name	Ayush Y Khanapure
Roll No.	2301083
Subject	Python

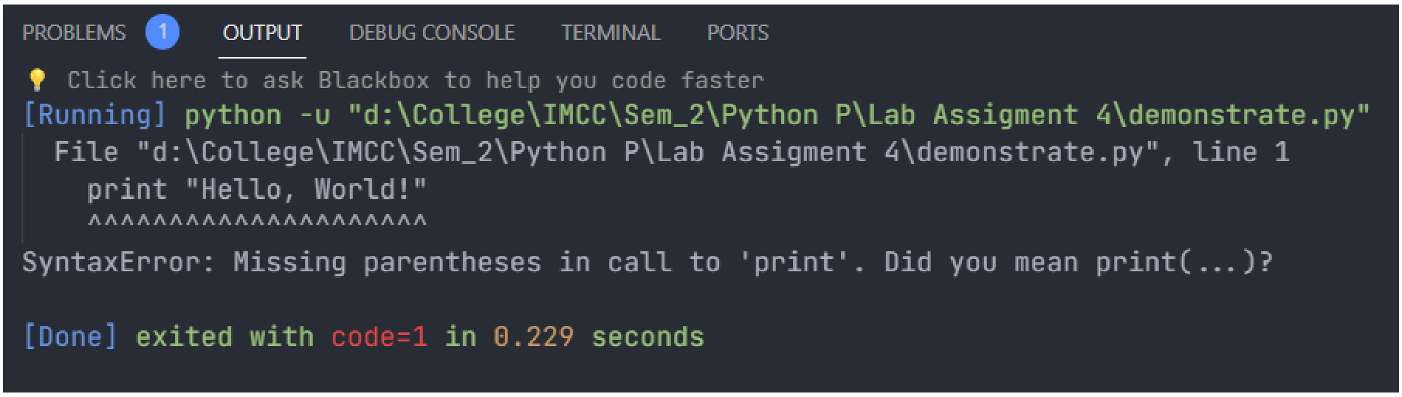
Assignment on Exception Handling and Files

Q1. Write Python program to demonstrate the following:

1. SyntaxError
Ans –

print "Hello, World!"

Output :

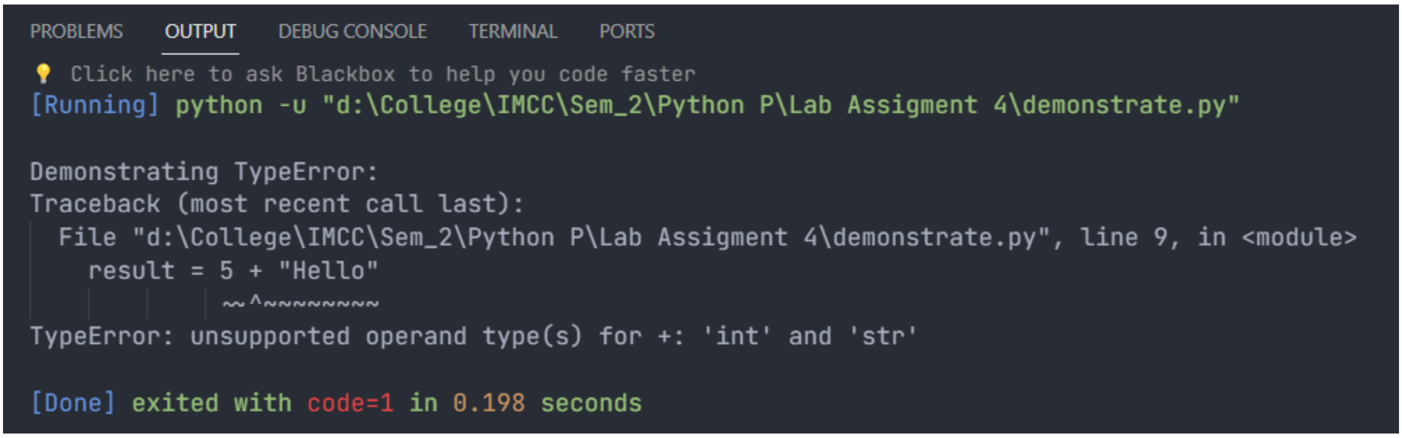


```
💡 Click here to ask Blackbox to help you code faster
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\demonstrate.py"
  File "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\demonstrate.py", line 1
    print "Hello, World!"
    ^^^^^^^^^^^^^^^^^^^^^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?

[Done] exited with code=1 in 0.229 seconds
```

2. TypeError
Ans –

result = 5 + "Hello"

Output :

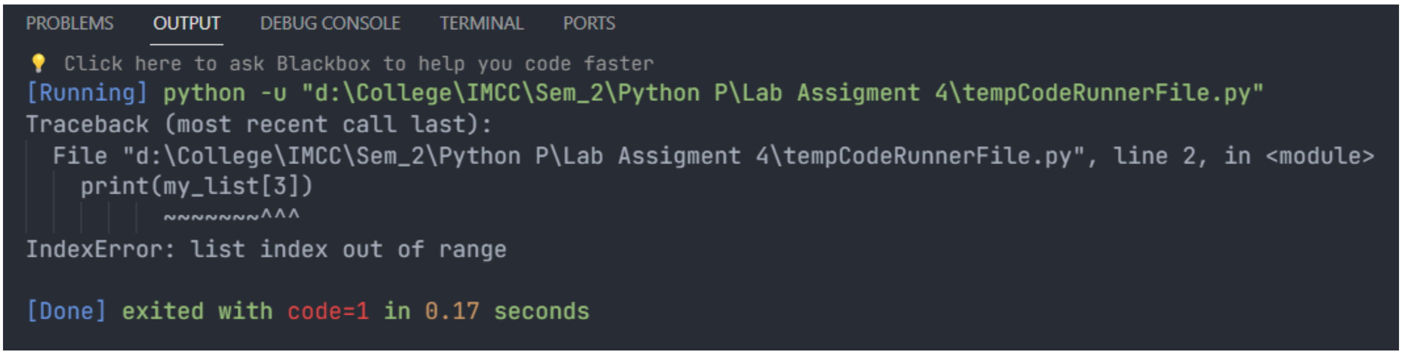


```
💡 Click here to ask Blackbox to help you code faster
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\demonstrate.py"

Demonstrating TypeError:
Traceback (most recent call last):
  File "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\demonstrate.py", line 9, in <module>
    result = 5 + "Hello"
             ~~^~~~~~~~~
TypeError: unsupported operand type(s) for +: 'int' and 'str'

[Done] exited with code=1 in 0.198 seconds
```

3. IndexError
Ans –

my_list = [1, 2, 3]
print(my_list[3])

<u>Output :</u>

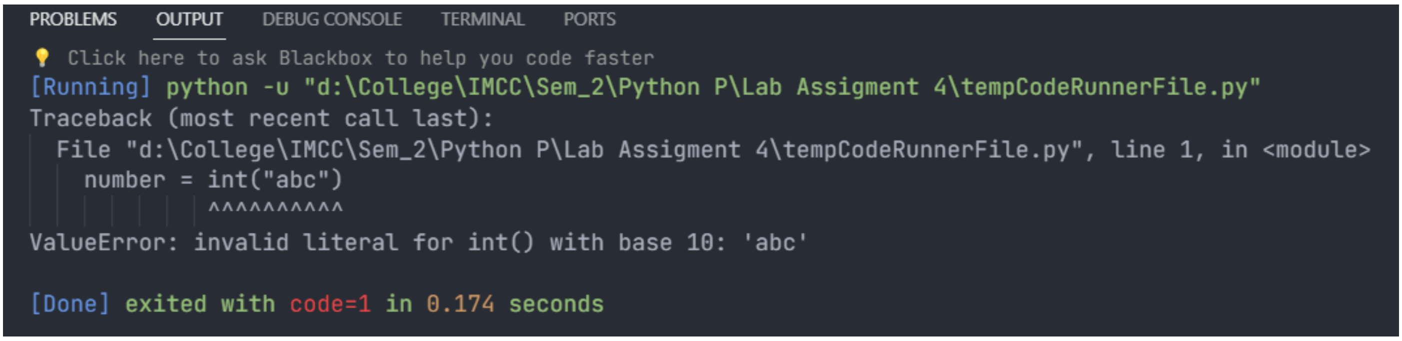


```
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\tempCodeRunnerFile.py"
Traceback (most recent call last):
  File "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\tempCodeRunnerFile.py", line 2, in <module>
    print(my_list[3])
          ~~~~~~~^^^
IndexError: list index out of range

[Done] exited with code=1 in 0.17 seconds
```

4. ValueError
Ans –

number = int("abc")

<u>Output :</u>

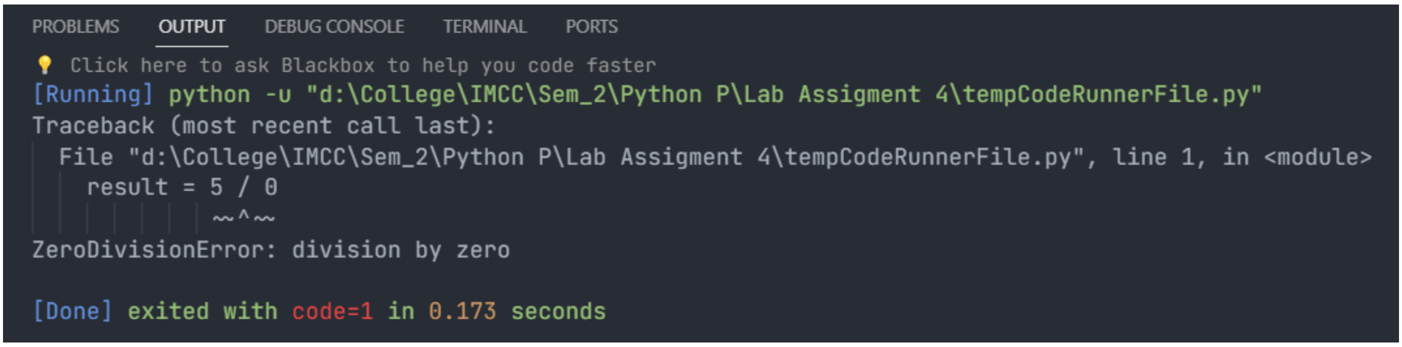


```
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\tempCodeRunnerFile.py"
Traceback (most recent call last):
  File "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\tempCodeRunnerFile.py", line 1, in <module>
    number = int("abc")
             ^^^^^^^^^^
ValueError: invalid literal for int() with base 10: 'abc'

[Done] exited with code=1 in 0.174 seconds
```

5. ZeroDivisionError
Ans –

result = 5 / 0

<u>Output :</u>



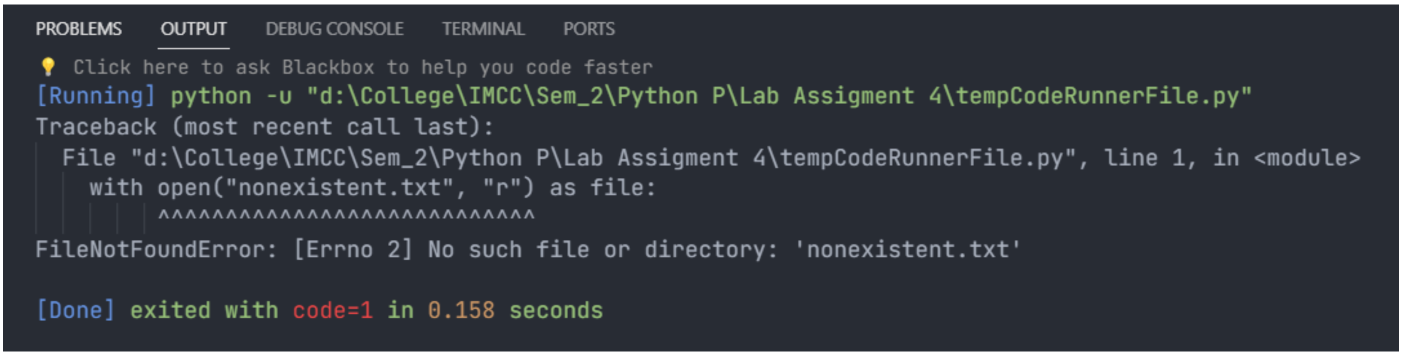
```
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\tempCodeRunnerFile.py"
Traceback (most recent call last):
  File "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\tempCodeRunnerFile.py", line 1, in <module>
    result = 5 / 0
             ~~^~~
ZeroDivisionError: division by zero

[Done] exited with code=1 in 0.173 seconds
```

6. fileNotFound

Ans –

```
with open("nonexistent.txt", "r") as file:
    contents = file.read()
```

Output :



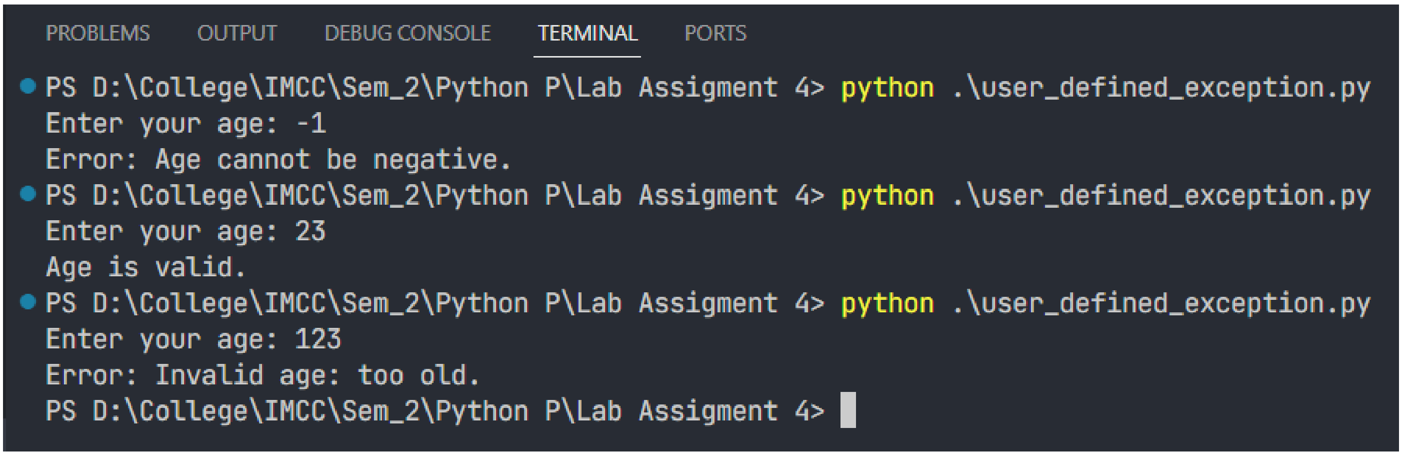
```
💡 Click here to ask Blackbox to help you code faster
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\tempCodeRunnerFile.py"
Traceback (most recent call last):
  File "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\tempCodeRunnerFile.py", line 1, in <module>
    with open("nonexistent.txt", "r") as file:
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'nonexistent.txt'

[Done] exited with code=1 in 0.158 seconds
```

2. Write Python program to raise user defined exception.

Ans –

```
class CustomError(Exception):
    def __init__(self, message):
        super().__init__(message)
        self.message = message

def validate_age(age):
    if age < 0:
        raise CustomError("Age cannot be negative.")
    elif age > 120:
        raise CustomError("Invalid age: too old.")
    else:
        print("Age is valid.")

try:
    age = int(input("Enter your age: "))
    validate_age(age)
except CustomError as e:
    print("Error:", e)
except ValueError:
    print("Invalid input: Please enter a valid integer.")
```

<u>Output :</u>

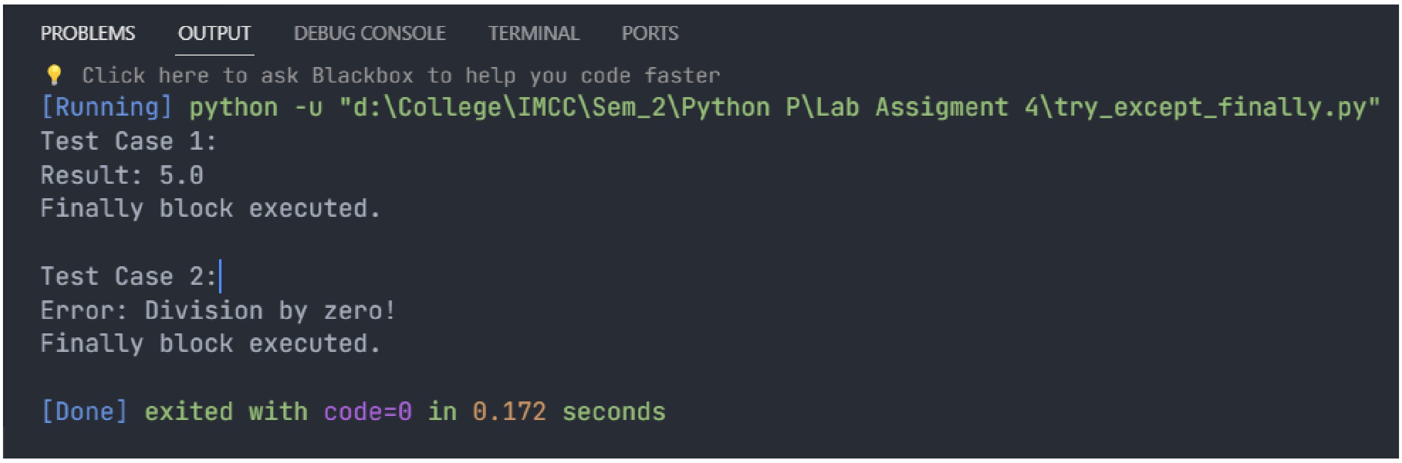


```
● PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 4> python .\user_defined_exception.py
  Enter your age: -1
  Error: Age cannot be negative.
● PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 4> python .\user_defined_exception.py
  Enter your age: 23
  Age is valid.
● PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 4> python .\user_defined_exception.py
  Enter your age: 123
  Error: Invalid age: too old.
  PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 4> 
```

3. Write Python program to demonstrate the use of try, except and finally block.
Ans –

```
def divide(x, y):
    try:
        result = x / y
    except ZeroDivisionError:
        print("Error: Division by zero!")
    else:
        print("Result:", result)
    finally:
        print("Finally block executed.")

print("Test Case 1:")
divide(10, 2)

print("\nTest Case 2:")
divide(10, 0)
```

<u>Output :</u>

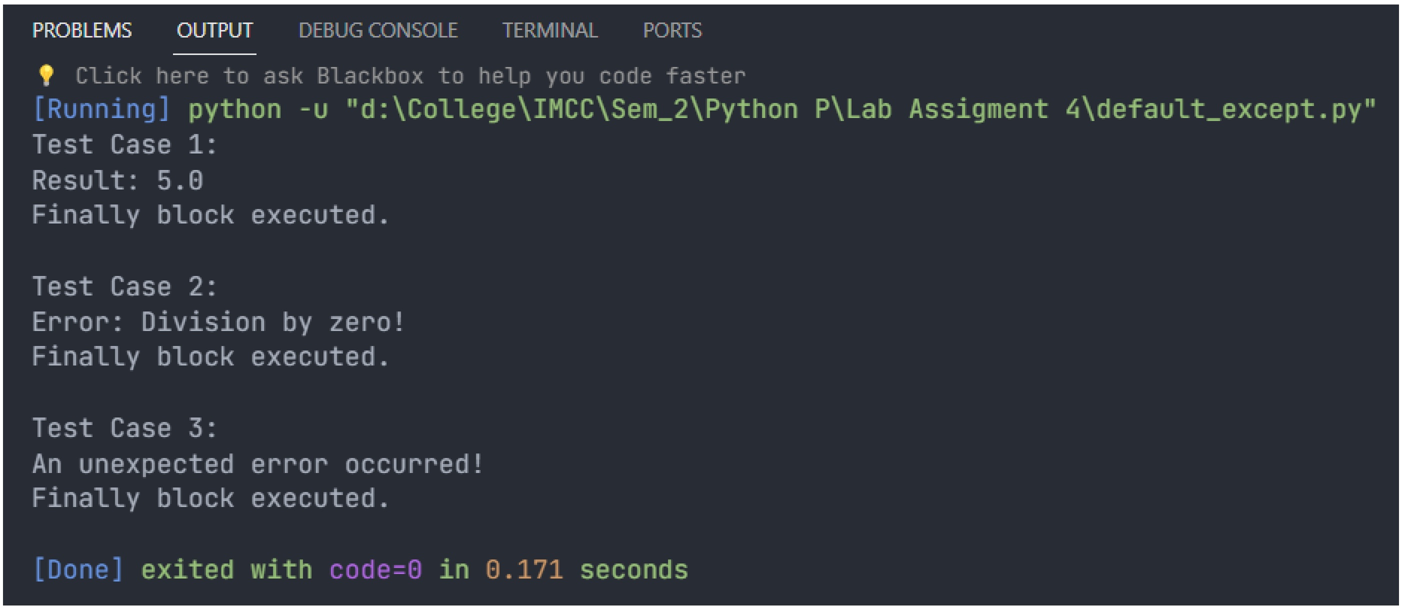


```
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\try_except_finally.py"
Test Case 1:
Result: 5.0
Finally block executed.

Test Case 2:
Error: Division by zero!
Finally block executed.

[Done] exited with code=0 in 0.172 seconds
```

4. Write Python program to demonstrate default except block.
Ans –

```
def divide(x, y):
    try:
        result = x / y
    except ZeroDivisionError:
        print("Error: Division by zero!")
    except ValueError:
        print("Error: Invalid input!")
    except:
        print("An unexpected error occurred!")
    else:
        print("Result:", result)
    finally:
        print("Finally block executed.")

print("Test Case 1:")
divide(10, 2)

print("\nTest Case 2:")
divide(10, 0)

print("\nTest Case 3:")
divide(10, 'a')
```

<u>Output :</u>

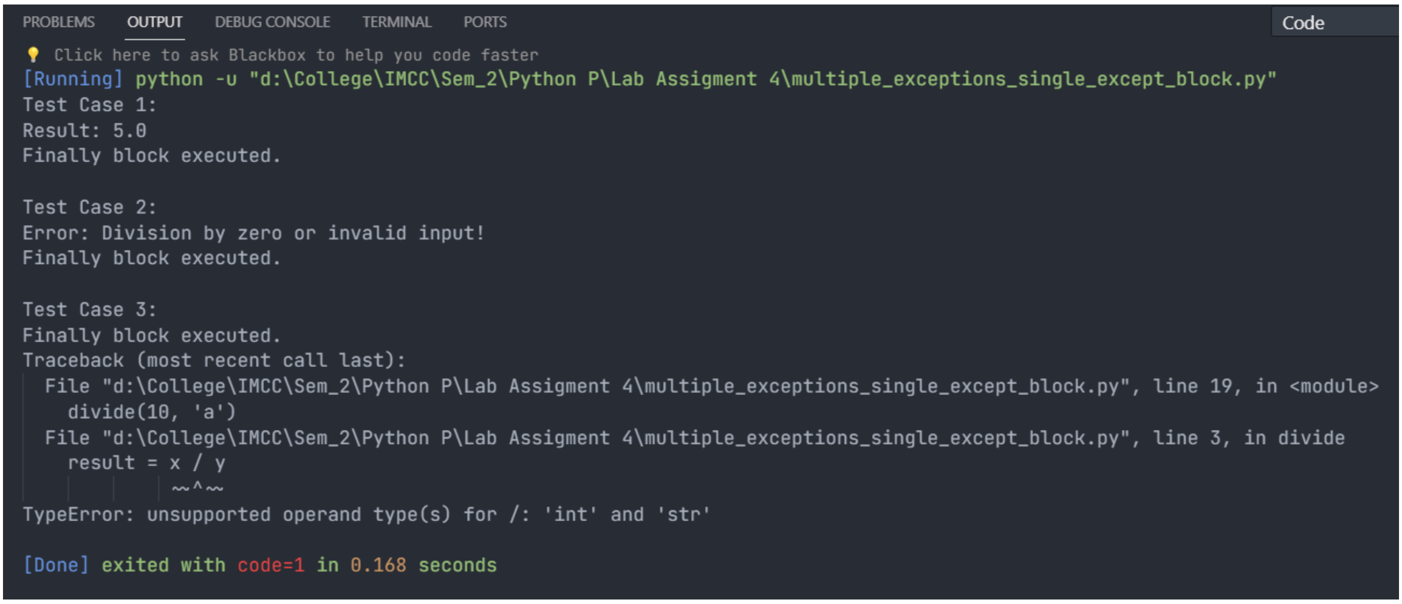


```
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\default_except.py"
Test Case 1:
Result: 5.0
Finally block executed.

Test Case 2:
Error: Division by zero!
Finally block executed.

Test Case 3:
An unexpected error occurred!
Finally block executed.

[Done] exited with code=0 in 0.171 seconds
```

5. Write Python program to handle multiple exceptions in single except block .
Ans –

```
def divide(x, y):
    try:
        result = x / y
    except (ZeroDivisionError, ValueError):
        print("Error: Division by zero or invalid input!")
    else:
        print("Result:", result)
    finally:
        print("Finally block executed.")

print("Test Case 1:")
divide(10, 2)

print("\nTest Case 2:")
divide(10, 0)

print("\nTest Case 3:")
divide(10, 'a')
```

Output :



```
💡 Click here to ask Blackbox to help you code faster
[Running] python -u "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\multiple_exceptions_single_except_block.py"
Test Case 1:
Result: 5.0
Finally block executed.

Test Case 2:
Error: Division by zero or invalid input!
Finally block executed.

Test Case 3:
Finally block executed.
Traceback (most recent call last):
  File "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\multiple_exceptions_single_except_block.py", line 19, in <module>
    divide(10, 'a')
  File "d:\College\IMCC\Sem_2\Python P\Lab Assigment 4\multiple_exceptions_single_except_block.py", line 3, in divide
    result = x / y
             ~~^~~
TypeError: unsupported operand type(s) for /: 'int' and 'str'

[Done] exited with code=1 in 0.168 seconds
```

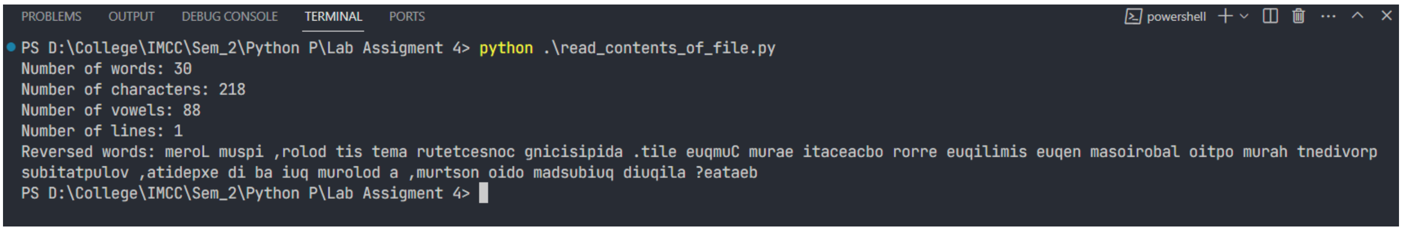
6. Write a program to read the contents of file and perform following operations

a) display number of words
b) display number of characters
c) display number of vowels
d) display number of lines
e) reverse each word and display it

Ans –

```
def count_words(file_content):
    words = file_content.split()
    return len(words)

def count_characters(file_content):
    return len(file_content)

def count_vowels(file_content):
    vowels = "aeiouAEIOU"
    return sum(1 for char in file_content if char in vowels)

def count_lines(file_content):
    return file_content.count('\n') + 1

def reverse_words(file_content):
    words = file_content.split()
    reversed_words = [word[::-1] for word in words]
    return ' '.join(reversed_words)

def main():

    try:
        with open("test.txt", 'r') as file:
            content = file.read()

            num_words = count_words(content)
            print("Number of words:", num_words)

            num_chars = count_characters(content)
            print("Number of characters:", num_chars)

            num_vowels = count_vowels(content)
            print("Number of vowels:", num_vowels)

            num_lines = count_lines(content)
            print("Number of lines:", num_lines)

            reversed_content = reverse_words(content)
            print("Reversed words:", reversed_content)

    except FileNotFoundError:
        print("Error: File not found.")

if __name__ == "__main__":
    main()
```

Output :



```
PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 4> python .\read_contents_of_file.py
Number of words: 30
Number of characters: 218
Number of vowels: 88
Number of lines: 1
Reversed words: meroL muspi ,rolod tis tema rutetcesnoc gnicisipida .tile euqmuC murae itaceacbo rorre euqilimis euqen masoirobal oitpo murah tnedivorp
subitatpulov ,atidepxe di ba iuq murolod a ,murtson oido madsubiuq diuqila ?eataeb
PS D:\College\IMCC\Sem_2\Python P\Lab Assigment 4> 
```