

PYTHON Lab Assignment4

Name: Rahul Kumar

Class: FYMCA (Div: B)

Roll No: 2201110

Assignment on Exception Handling:

A. Write Python program to demonstrate the following:

1. SyntaxError

```
print(5;)
```

```
File "main.py", line 1
print(5;)
SyntaxError: invalid syntax
> █
```

2. TypeError

```
x = "5"
if x > 5:
    print(x)
```

```
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    if x > 5:
TypeError: '>' not supported between instances of 'str' and 'int'
> █
```

3. IndexError

```
list = [71,2,53,4]
print(list[5])
```

```
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    print(list[5])
IndexError: list index out of range
> █
```

4. ValueError

```
num = int(input("Enter an integer: "))
print(num)
```

```
Enter an integer: f
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    num = int(input("Enter an integer: "))
ValueError: invalid literal for int() with base 10: 'f'
> █
```

5. ZeroDivisionError

```
a = 5
b = 0
print(a/b)
```

```
Traceback (most recent call last):
  File "main.py", line 3, in <module>
    print(a/b)
ZeroDivisionError: division by zero
> █
```

6. FileNotFoundError

```
with open('nofile.txt') as file:
    print(file.read())
```

```
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    with open('nofile.txt') as file:
FileNotFoundError: [Errno 2] No such file or directory: 'nofile.txt'
> █
```

PYTHON Lab Assignment4

B. Write Python program to raise user defined exception

User-defined exception class

```
class MyException(Exception):  
    pass
```

Function that raises the exception

```
def raise_exception():  
    raise MyException("Custom exception!")
```

Main program

```
try:  
    raise_exception()  
except MyException as e:  
    print("Custom exception caught:", str(e))
```

```
Custom exception caught: Custom exception!  
❖ □
```

C. Write Python program to demonstrate the use of try, except and finally block

try:

```
    file = open('nofile.txt')  
    print(file.read())
```

except FileNotFoundError:

```
    print("file not available")
```

finally:

```
    file.close()
```

```
file not available  
Traceback (most recent call last):  
  File "main.py", line 7, in <module>  
    file.close()  
NameError: name 'file' is not defined. Did you mean: 'filter'?  
❖ □
```

D. Write Python program to demonstrate default except block

try:

```
    with open('nofile.txt') as file:  
        print(file.read())
```

except:

```
    print("file not available")
```

```
file not available  
Traceback (most recent call last):  
  File "main.py", line 7, in <module>  
    file.close()  
NameError: name 'file' is not defined. Did you mean: 'filter'?  
❖ □
```

E. Write Python program to handle multiple exceptions in single except block

try:

```
    a = 10 / 10  
    b = int('abc')  
    c = [1, 2, 3]  
    print(c[4])
```

except (ZeroDivisionError, ValueError, IndexError) as e:

```
    print(f"An exception occurred: {type(e).__name__}")
```

```
An exception occurred: ValueError  
❖ □
```