Name : Neha Dumane

Roll No : 2201046

A] Write Python program to demonstrate the following:

1. SyntaxError
2. TypeError
3. IndexError
4. ValueError
5. ZeroDivisionError
6. fileNotFound

Code :

```
//SyntaxError
len('hello') = 5
//TypeError
geek = "GeeksforGeeks"
print(geek())
//IndexError
languages = ['Python', 'JavaScript', 'Java']
print(languages[3])
//ValueError
```

```
import math

math.sqrt(-10)

//ZeroDivisionError

x = 5

y = 0

z = x/y

print(z)

//fileNotFoundError

fo = open("myfile.txt","r")

print("File opened")
```

Output :

```
SyntaxError: cannot assign to function call here. Maybe you meant '==' instead of '='?

TypeError: 'str' object is not callable

IndexError: list index out of range

ValueError: math domain error

ZeroDivisionError: division by zero

FileNotFoundError: [Errno 2] No such file or directory: 'myfile.txt'
```

B] Write Python program to raise user defined exception

Code :

```
class InvalidAgeException(Exception):
    "Raised when the input value is less than 18"
    pass
number = 18
try:
    input_num = int(input("Enter a number: "))
    if input_num < number:
        raise InvalidAgeException
    else:
        print("Eligible to Vote")
except InvalidAgeException:
    print("Exception occurred: Invalid Age")
```

Output :

```
Enter a number: 15
Exception occurred: Invalid Age
```

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

Code :

```
def divide(x, y):
    try:
        result = x // y
    except ZeroDivisionError:
        print("Sorry ! You are dividing by zero ")
    else:
        print("Yeah ! Your answer is :", result)
    finally:
        print('This is always executed')
divide(3, 0)
```

Output :

```
Sorry ! You are dividing by zero
This is always executed
```

D] Write Python program to demonstrate default except block

Code :

```
try:
    numerator = 10
    denominator = 0
    result = numerator/denominator
    print(result)
except:
    print("Error: Denominator cannot be 0.")
```

Output :

```
Error: Denominator cannot be 0.
```

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

Code :

```
string = input()
try:
    num = int(input())
    print(string+num)
except (TypeError, ValueError) as e:
    print(e)
```

Output :

```
a

h

invalid literal for int() with base 10: 'h'
```