PYTHON

ASSIGNMENT 7

STUDENT NAME: TELORE GANESH BHASKAR

CLASS: TYBBACA

ROLLNO: 235353

GUIDE: PROF. LANDE R.D

ASSIGNMENT BASED ON: Exception Handling

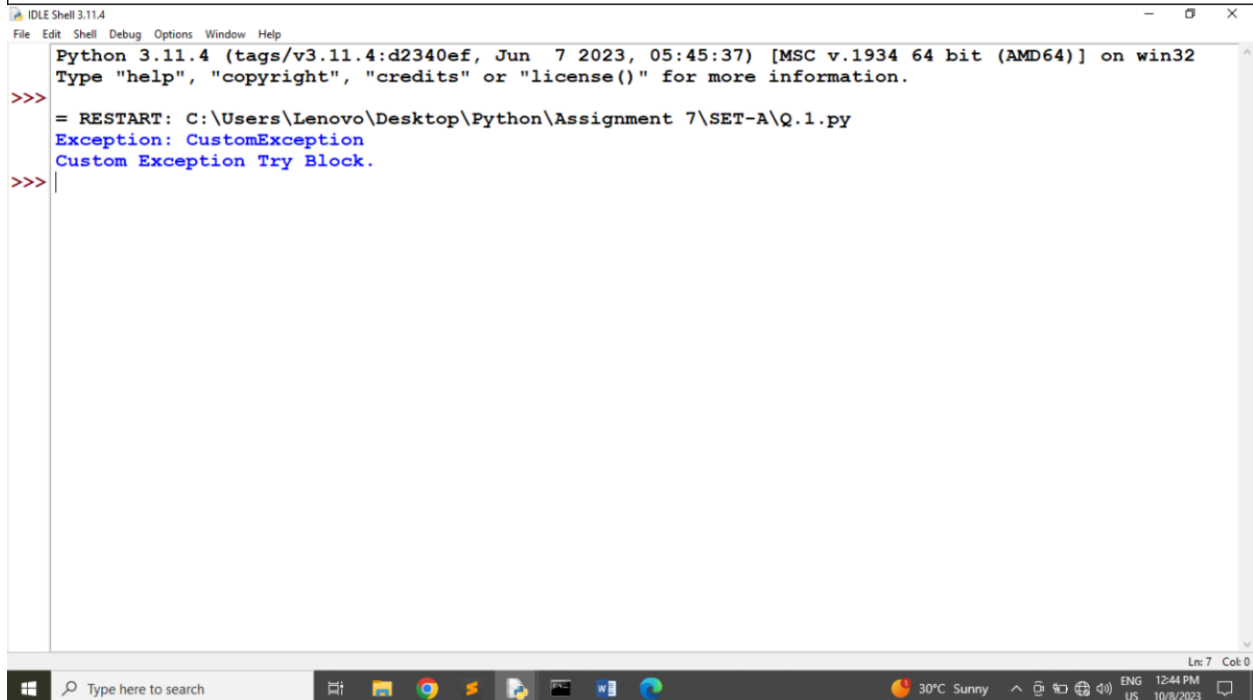| ASSIGNMENT 7 |
|---|
| **SET-A** |

Q.1 Define a custom exception class which takes a string message as attribute.

Ans:
```
class CustomException(Exception):
    def __init__(self,msg):
        self.msg=msg
try:
    raise CustomException("\nCustom Exception Try Block.")

except CustomException as e:
    print("Exception: CustomException ",e.msg)
```

Output:

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-A\Q.1.py
Exception: CustomException
Custom Exception Try Block.
>>>
```

ASSIGNMENT 7
SET-A

Q.2 Write a function called oops that explicitly raises a IndexError exception when called. Then write another function that calls oops inside a try/except statement to catch the error.
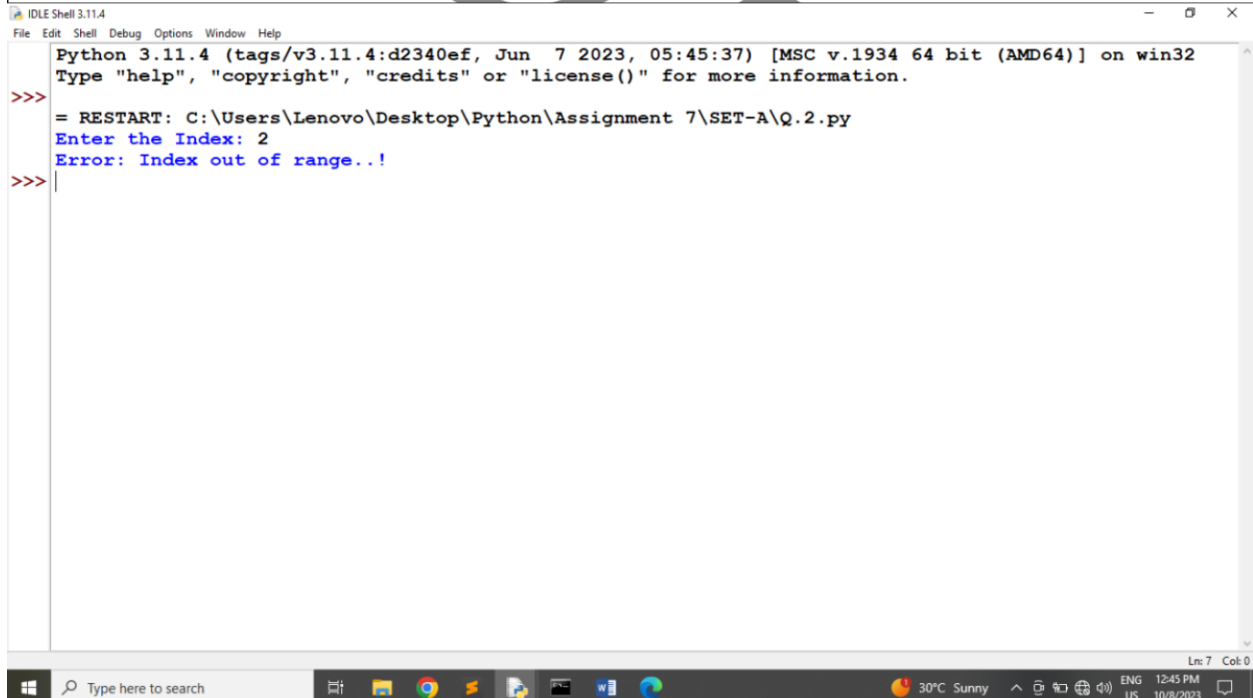
Ans:
```python
def oop():
    a=[1,2]
    n=int(input("Enter the Index: "))
    if(n>1):
        raise IndexError
    else:
        print(f"Value is: {a[n]}")

def newoop():
    try:
        oop()
    except IndexError:
        print("Error: Index out of range..!")

newoop()
```

Output:

IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-A\Q.2.py
Enter the Index: 2
Error: Index out of range..!
>>>
```
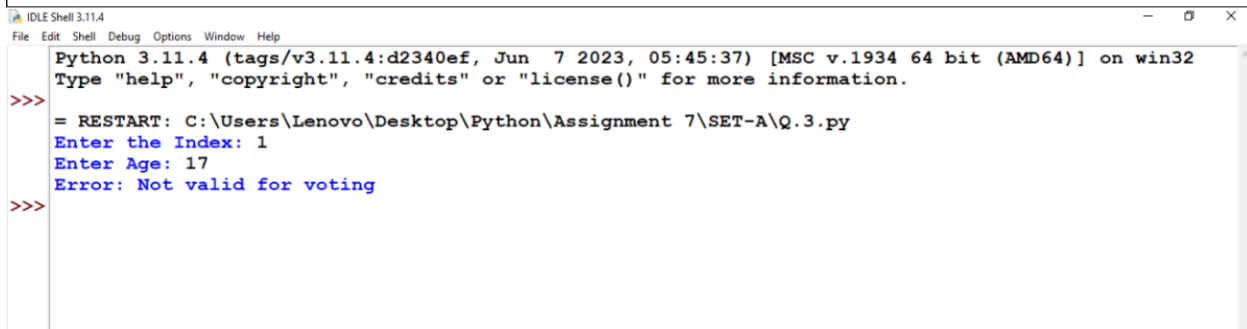
ASSIGNMENT 7
SET-A

Q.3 Change the oops function you just wrote to raise an exception you define yourself, called MyError, and pass an extra data item along with the exception. Then, extend the try statement in the catcher function to catch this exception and its data in addition to IndexError, and print the extra data item.

Ans:
```python
class Error(Exception):
    pass
class MyError(Error):
    pass
def oop():
    a=[1,2]
    n=int(input("Enter the Index: "))
    string1=int(input("Enter Age: "))
    if(n>1):
        raise IndexError
    if(string1<18):
        raise MyError
    else:
        print(a[n])
        print("Valid for voting")
def newoop():
    try:
        oop()
    except IndexError:
        print("Error: Index out of range..!")
    except MyError:
        print("Error: Not valid for voting")

newoop()
```

Output:

```
IDLE Shell 3.11.4                                                    —   □   ×
File  Edit  Shell  Debug  Options  Window  Help
    Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    = RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-A\Q.3.py
    Enter the Index: 1
    Enter Age: 17
    Error: Not valid for voting
>>>
```

ASSIGNMENT 7
SET-B

Q.1 Define a class Date(Day, Month, Year) with functions to accept and display it. Accept date from user. Throw user defined exception "invalidDateException" if the date is invalid.

Ans:
```python
class Date:
    def getdate(self):
        print("Note: Fill data only in Number Format")
        self.day=int(input("enter day: "))
        self.month=int(input("enter month: "))
        self.year=int(input("enter year: "))
    def show(self):
        try:
            if(self.day>31):
                raise Exception
            if(self.month>12):
                raise Exception
            if(self.year<0000):
                raise Exception
            print(f"{self.day}/{self.month}/{self.year}")
        except Exception:
            print("invalid date")

obj=Date()
obj.getdate()
obj.show()
```
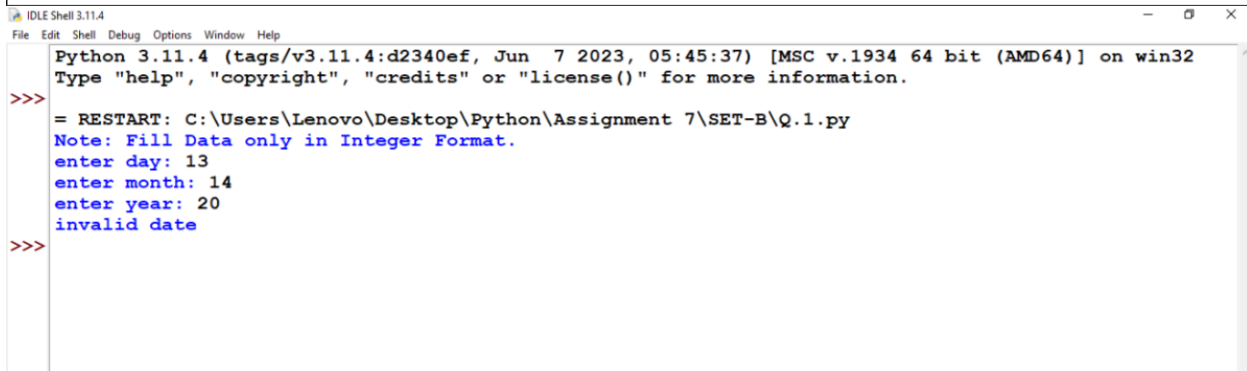
Output:


```
IDLE Shell 3.11.4                                                    —  ☐  ✕
File  Edit  Shell  Debug  Options  Window  Help
    Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    = RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-B\Q.1.py
    Note: Fill Data only in Integer Format.
    enter day: 13
    enter month: 14
    enter year: 20
    invalid date
>>>
```

ASSIGNMENT 7
SET-B

Q.2 Write text file named test.txt that contains integers, characters and float numbers. Write a Python program to read the test.txt file. And print appropriate message using exception

Ans:
```
'''file=open("test.txt","w")
L=["I am GT. \n and I am 19 years old.\n i am 5ft 6.6 inches tall."]
file.write("Hello There \n")
file.writelines(L)'''
try:
    file=open("test2.txt", "r")
    print(file.read())
    file.close()
except:
    print("unable to open file beacuse file is not created or file does not contain any data")
```

Output:

OP1

```
IDLE Shell 3.11.4                                                      —  □  ×
File  Edit  Shell  Debug  Options  Window  Help
    Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    = RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-B\Q.2.py
    unable to open file beacuse file is not created or file does not contain any data
>>>
```

OP2

```
IDLE Shell 3.11.4                                                      —  □  ×
File  Edit  Shell  Debug  Options  Window  Help
    Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    = RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-B\Q.2.py
    unable to open file beacuse file is not created or file does not contain any data
>>>
    ================ RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-B\Q.2.py ================
    Hello There
    I am GT.
     and I am 19 years old.
     i am 5ft 6.6 inches tall.
>>>
```

ASSIGNMENT 7
SET-C

Q.1 Write a function called safe (func, *args) that runs any function using apply, catches any exception raised while the function runs, and prints the exception using the exc_type and exc_value attributes in the sys module. Then, use your safe function to run the oops function you wrote in Exercises 3. Put safe in a module file called tools.py, and pass it the oops function interactively. Finally, expand safe to also print a Python stack trace when an error occurs by calling the built-in print_exc() function in the standard traceback module (see the Python library reference manual or other Python books for details)

Ans:

program1:
```
import traceback
def safe(num):
    try:
        op=num/0
    except Exception as e:
        exc_type=type(e).__name__
        exc_value=str(e)
        print("Exception Type:",exc_type)
        print("Exception Value: ",exc_value)
        traceback.print_exc()
```
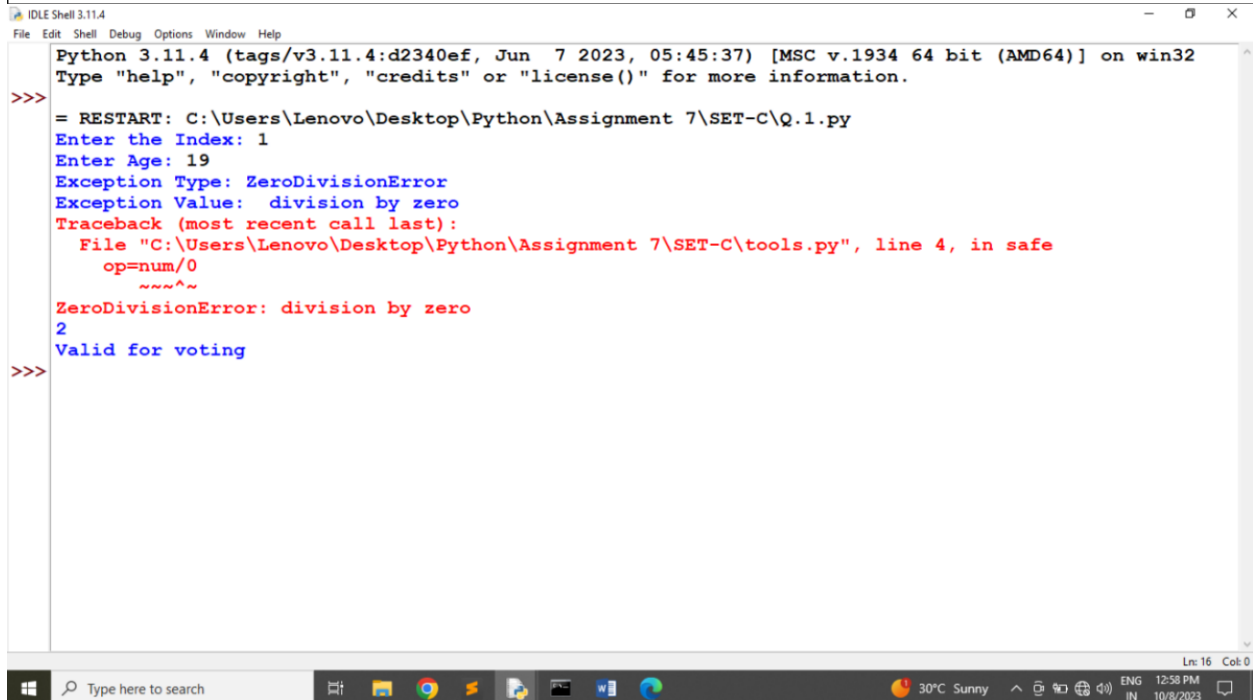
program2:
```
class Error(Exception):
    pass
class MyError(Error):
    pass
import tools
def oop():
    a=[1,2]
    n=int(input("Enter the Index: "))
    string1=int(input("Enter Age: "))
    tools.safe(5)
    if(n>1):
        raise IndexError
    if(string1<18):
        raise MyError
    else:
        print(a[n])
        print("Valid for voting")
```

```
def newoop():
    try:
        oop()
    except IndexError:
        print("Error: Index out of range..!")
    except MyError:
        print("Error: Not valid for voting")


newoop()
```

Output:

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-C\Q.1.py
Enter the Index: 1
Enter Age: 19
Exception Type: ZeroDivisionError
Exception Value:  division by zero
Traceback (most recent call last):
  File "C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-C\tools.py", line 4, in safe
    op=num/0
       ~~~^~
ZeroDivisionError: division by zero
2
Valid for voting
>>>
```
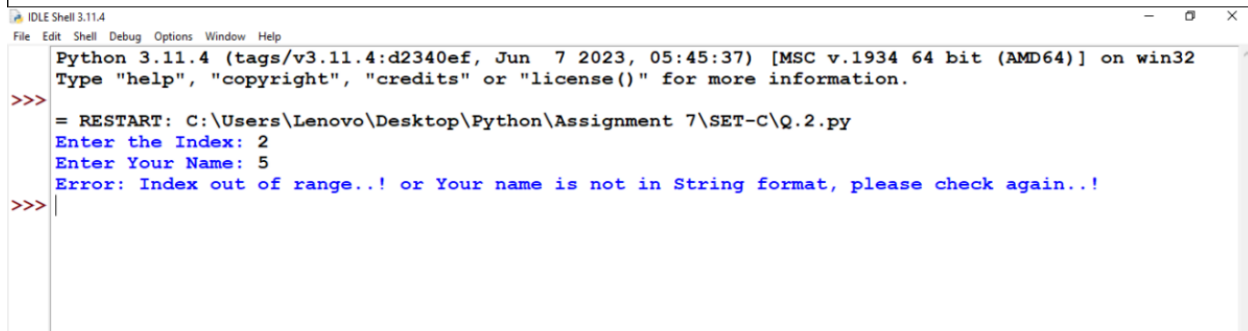
ASSIGNMENT 7
SET-C

Q.2 Change the oops function in question 4 from SET A to raise an exception you define yourself, called MyError, and pass an extra data item along with the exception. You may identify your exception with either a string or a class. Then, extend the try statement in the catcher function to catch this exception and its data in addition to IndexError, and print the extra data item. Finally, if you used a string for your exception, go back and change it be a class instance.

Ans:

```
class Error(Exception):
    pass
class MyError(Error):
    pass
def oop():
    a=[1,2]
    n=int(input("Enter the Index: "))
    name=input("Enter Your Name: ")
    if(n>1):
        raise IndexError
    if name.isdigit():
        raise MyError
    else:
        print(a[n])
        print("Your Name is: "+name)
def newoop():
    try:
        oop()
    except (IndexError,MyError):
        print("Error: Index out of range..! or Your name is not in String format, please check again..!")

newoop()
```

Output:


```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Lenovo\Desktop\Python\Assignment 7\SET-C\Q.2.py
Enter the Index: 2
Enter Your Name: 5
Error: Index out of range..! or Your name is not in String format, please check again..!
>>>
```