Exercise.

Create a Name Generator App using Function and Random randint() method.

firstname = [“John”,”Mark”,”Chris”,”James”,”Philip”]

1. Create 3 lists for first name, middle name, and last name with 5 items per list
2. The application will ask the user to generate a new name.
   Example:
   Do you want to generate a new name? [y/n]:
3. If yes, use a random number 0 - 4 to randomly select the items in the lists
4. Display the generated name
   “Your new name is __________________”
   and repeat the input statement in step 2.
5. If no, display “Thank you!” on the screen and display all the names that the user generated.

Example Run:

```
Do you want to generate a new name? y
Your new name is Juan Cruz Reyes

Do you want to generate a new name? y
Your new name is John Doe Gates

Do you want to generate a new name? n
Thank you!
List of generated names:
Juan Cruz Reyes
John Doe Gates
```

Requirements:

- **Function**
- **Random**
- **Python List**
- **while loop**
- **If statement**

Exception

- An event which occurs during the execution of a program that disrupts the normal flow of the program’s instructions.

Examples:

○ Division by zero

○ Invalid input

○ File not found

`try` `except` `finally`

Types of Exception

-Checked

- Checked exceptions are checked at compile-time.
- Exceptions are "checked" because they are subject to the "**catch**" or "**specify requirement**" otherwise, the program code will not compile.

Examples: Invalid Syntax, Incorrect statements

-Unchecked (Runtime)

- Unchecked exceptions are not checked at compile time.
- Errors are not subject to the "**catch**" or "**specify requirement**".
- It occurs during Runtime

Examples: invalid input, invalid arithmetic operations, number divided by zero

Handling Unchecked exception in Python

- try is used to test a block of code for errors
- except is used to handle errors
- finally is used to execute block of code regardless of the result of the try and except blocks

Using try…except

```
print(age)
```

Output:

```
Traceback (most recent call last):
  File "~~F:\DICTPython\Exception.py~~", line 1, in <module>
    print(age)
NameError: name 'age' is not defined
```

```
try:
    print(age)
except:
    print(“Age variable is not defined")
    print(“You must assign a value first or declare it")
```

Output:

```
Age variable is not defined
You must assign a value first or declare it
```

Print a specific error message if the try block raises a NameError

```
try:
    print(age)
except NameError:
    print("Variable age is not defined")
except:
    print("Something else went wrong")
```

Output:

```
Variable age is not defined
```

This statement will be executed since the try block encountered a NameError

Else

else is used to define a block of code to be executed if no errors were raised

```
try:
    print("Hello World")
except:
    print("An error occured")
else:
    print("No error")
```

Output:

```
Hello World
No error
```

This statement will be executed since there’s no error inside the try block

Finally

finally block will be executed regardless if the try block raises an error or not.

```
try:
    print("Hello World")
except:
    print("An error occurred")
else:
    print("No error")
finally:
    print(“Finished")
```

```
Output:
Hello World
No error
Finished
```

```
try:
    print(age)
except:
    print("An error occurred")
else:
    print("No error")
finally:
    print(“Finished")
```

```
Output:
An error occurred
Finished
```

Raise an Exception

raise is used to throw an exception if a condition occurs.

```
age = -1
if age < 0:
  raise Exception("Sorry, age is out of range")
```

```
age = "30"
if not type(age) is int:
  raise TypeError("Can accept integer value only")
```

Python Unchecked Exceptions

SyntaxError

```
a = 1
if a < 0
   print("Negative")
```

ZeroDivisionError

```
num = 30
result = num/0
print(result)
```

NameError

```
print(result)
```

TypeError

```
num = “five”
result = num+5
print(result)
```

ValueError

```
num = int(input("Enter a number: "))
result = num+2
print(result)
```

Sample input: two

Practical

Create a Simple App Calculator

1. The application will ask the user to choose one of the four math operations (Addition, Subtraction, Multiplication and Division)
2. The application will ask the user for two numbers
3. Display the result
4. The application will ask if the user wants to try again or not.
5. If yes, repeat Step 1.
6. If no, Display “Thank you!” and the program will exit
7. Use appropriate Exceptions to capture errors during runtime.

Python File Handling

Allows the user to read, write, update, and delete files

open() – used to open a file and returns a file object

close() – used to close a file

read(), readline() – used to read content of the file

write() – used to write to an existing file

delete() – used to remove files

Open File

filename or file path

mode

```
file = open("file.txt", [r,a,w,x])
```

"r" (Read) - Opens a file for reading, error if the file does not exist

"a" (Append) - Opens a file for appending, creates the file if it does not exist

"w" (Write) - Opens a file for writing, creates the file if it does not exist

"x" (Create) - Creates the specified file, returns an error if the file exists

Python File Reading

```
try:
    fr = open("file.txt","r")
    content = fr.read()
    print("Content:")
    print(content)
    fr.close()
except FileNotFoundError:
    print("File is not found")
```

```python
for line in file:
    line = line.strip("\n")
    list = line.split("#")
    print(f" {list[0]}\t{list[1]} ")
```

data.txt

```
Chris#0950
John#0910
James#0930
```

Python File Writing

```
try:
        fw = open("file.txt","w")
        content="Hello World“
        fw.write(content)
        fw.close()
except FileNotFoundError:
       print("File is not found")
```

Python File Writing

```
try:
    fw = open("file.txt",“a")
    content="Hello World“
    fw.write(content)
    fw.close()
except FileNotFoundError:
    print("File is not found")
```

Python file reading using for loop and strip() function

```
try:
        fr = open("file.txt","r")
        print("Content:")
        for line in fr:
            line = line.strip(“\n”)
            print(line)
        fr.close()
except FileNotFoundError:
      print("File is not found")
```

file.txt contains the following :

```
Hello World
Hello Universe
Hello Philippines
Hello Iloilo City
```

Output of the Program:

```
Hello World
Hello Universe
Hello Philippines
Hello Iloilo
```

Python file reading using for loop, array, strip() and split() functions

```
try:
    arr = []
    fr = open("file.txt","r")
    print("Content:")
    for line in fr:
        line = line.strip("\n")
        arr = line.split("#")
        print(f"{arr[0]} {arr[1]} {arr[2]}")
    fr.close()
except FileNotFoundError:
    print("File is not found")
```

Example: file.txt contains the following:

```
Juan Dela Cruz#Iloilo City#30
John Mark#Antique#25
Bill Gates#Guimaras#23
```

Output of the Program:

```
Juan Dela Cruz Iloilo City 30
John Mark Antique 25
Bill Gates Guimaras 23
```

Python file reading using for loop, list, strip() and split() functions

```
try:
    mylist = list()
    a = list()
    fr = open("file.txt","r")
    print("Content:")
    for line in fr:
        line = line.strip("\n")
        a = line.split("#")
        mylist.append(a)
        print(f"{mylist[0]} {mylist[1]} {mylist[2]}")

    print(mylist[2][1])
    fr.close()
except FileNotFoundError:
    print("File is not found")
```

Example: file.txt contains the following:

```
Juan Dela Cruz#Iloilo City#30
John Mark#Antique#25
Bill Gates#Guimaras#23
```

Output of the Program:

```
Juan Dela Cruz    Iloilo City    30
John Mark          Antique       25
Bill Gates         Guimaras       23
```

Delete a File

To delete a file, you must import the OS module, and run its `os.remove()` function:

```
import os
os.remove("file.txt")
```

Check if File exist:

To avoid getting an error, you might want to check if the file exists before you try to delete it:

Check if file exists, *then* delete it:

```
if os.path.exists("file.txt"):
    os.remove("file.txt")
else:
    print("The file does not exist")
```

Delete Folder

To delete an entire folder, use the `os.rmdir()` method:

```
import os
os.rmdir("myfolder")
```

Assignment 1

Create a program that reads a sentence inside a textfile with a filename data.txt.
The data.txt should contain the following:

Python Programming Essential Course

the quick brown fox jumps over a lazy dog

The program should display the sentences in uppercase on the screen.
The program should also display the total numbers of words on the screen.

Example Input/Output of the Program:

```
PYTHON PROGRAMMING ESSENTIAL COURSE
Total number of words: 4
---------------------------------------------------------
THE QUICK BROWN FOX JUMPS OVER A LAZY DOG
Total number of words: 9
```

Assignment 2

ATM PROGRAM (Apply function, exception and file handling)

[1]. Check Balance

[2]. Deposit

[3]. Withdraw

[4]. Exit

Assignment 3

Create a JackNPoy game (Computer VS User) using Function and Random randint() method.

The program should accept input from the user, then the program should generate computer bet using randint().
The program should determine the winner and display the computer bet for verification of the winner.

Example Run:

```
Jack N Poy Game
Legend:
P/p – Paper
R/r – Rock
S/s – Scissors

Enter user bet: R
Computer wins!
Paper covers rock
Computer bet: P
R vs P
Winner: P
```