

Task 9

Exercise 10.1: Learning Python.

Open a blank file in your text editor and write a few lines summarizing what you've learned about Python so far. Start each line with the phrase In Python you can... Save the file as learning_python.txt in the same directory as your exercises from this chapter. Write a program that reads the file and prints what you wrote three times. Print the contents once by reading in the entire file, once by looping over the file object, and once by storing the lines in a list and then working with them outside the with block.

```
In [ ]: fileName="learning_python.txt"

print("Reading the entire file:\n")
with open(fileName) as f:
    lines=f.read()
print(lines)

print("\nReading the file line by line:\n")
with open(fileName) as f:
    for line in f:
        print(line.rstrip())

print("\nReading the file line by line and storing it in a list:\n")
with open(fileName) as f:
    lines=f.readlines()

for line in lines:
    print(line.rstrip())
```

Reading the entire file:

```
In Python you can have dynamic lists  
In Python you can do almost anything you want very intuitively  
In Python you can write 5 lines of code to do something that would take 50 lines o  
f code in another language
```

Reading the file line by line:

```
In Python you can have dynamic lists  
In Python you can do almost anything you want very intuitively  
In Python you can write 5 lines of code to do something that would take 50 lines o  
f code in another language
```

Reading the file line by line and storing it in a list:

```
In Python you can have dynamic lists  
In Python you can do almost anything you want very intuitively  
In Python you can write 5 lines of code to do something that would take 50 lines o  
f code in another language
```

Exercise 10.2: Learning C.

You can use the replace() method to replace any word in a string with a different word. Here's a quick example showing how to replace 'dog' with 'cat' in a sentence:

```
>>> message = "I really like dogs."  
>>> message.replace('dog', 'cat')  
'I really like cats.'
```

Read in each line from the file you just created, learning_python.txt, and replace the word Python with the name of another language, such as C. Print each modified line to the screen.

```
In [ ]: fileName="learning_python.txt"  
  
with open(fileName) as f:  
    lines=f.readlines()  
  
for line in lines:  
    print(line.replace('Python','C').rstrip())
```

```
In C you can have dynamic lists  
In C you can do almost anything you want very intuitively  
In C you can write 5 lines of code to do something that would take 50 lines of cod  
e in another language
```

Exercise 10.3: Guest.

Write a program that prompts the user for their name. When they respond, write their name to a file called guest.txt.

```
In [ ]: name=input("Enter your name: ")

fileName="guest.txt"

with open(fileName, 'w') as f:
    f.write(name)

print("Testing if the name's been written in the file:\n")
with open(fileName) as f:
    lines=f.read()
print(lines)
```

Testing if the name's been written in the file:

Muid

Exercise 10.4: Guest Book.

Write a `while` loop that prompts users for their name. When they enter their name, print a greeting to the screen and add a line recording their visit in a file called `guest_book.txt`. Make sure each entry appears on a new line in the file.

```
In [ ]: fileName="guest_book.txt"

print("Enter 'q' to quit the program.\n")
while True:
    name=input("\nEnter your name: ")
    if name=='q':
        break
    else:
        with open(fileName, 'a') as f:
            f.write(f"\n{name}")
        print(f"Hi {name.title()}, your name has been added to the file.")

print("\nTesting if the name's been written in the file:\n")
with open(fileName) as f:
    lines=f.read()
print(lines)
```

Enter 'q' to quit the program.

Hi Muid, your name has been added to the file.
Hi Ahmed, your name has been added to the file.
Hi Ali, your name has been added to the file.
Hi Hamza, your name has been added to the file.

Testing if the name's been written in the file:

muid
ahmed
ali
hamza

Exercise 10.5: Programming Poll.

Write a `while` loop that asks people why they like programming. Each time someone enters a reason, add their reason to a file that stores all the responses.

```
In [ ]: fileName="programming_poll.txt"

responses=[]
while True:
    response=input("\nWhy do you like programming? ")
    responses.append(response)
    repeat=input("Would you like to let another person respond? (yes/no) ")
    if repeat=='no':
        break

with open(fileName, 'w') as f:
    for response in responses:
        f.write(f"{response}\n")

print("\nTesting if the responses have been written in the file:\n")
with open(fileName) as f:
    lines=f.read()
print(lines)
```

Testing if the responses have been written in the file:

```
It's fun
It helps me create things
```

Exercise 10.6: Addition.

One common problem when prompting for numerical input occurs when people provide text instead of numbers. When you try to convert the input to an `int`, you'll get a `ValueError`. Write a program that prompts for two numbers. Add them together and print the result. Catch the `ValueError` if either input value is not a number, and print a friendly error message. Test your program by entering two numbers and then by entering some text instead of a number.

Program with input 3 and 5:

```
In [ ]: try:
    x=int(input("Enter a number: "))
    x=int(x)

    y=input("Enter another number: ")
    y=int(y)
except ValueError:
    print("You didn't enter a number.")
```

```
    else:  
        print(f"{x} + {y} = {x+y}")
```

3 + 2 = 5

Program with input 3 and 'five':

```
In [ ]: try:  
    x = int(input("Enter a number: "))  
    x = int(x)  
  
    y = input("Enter another number: ")  
    y = int(y)  
except ValueError:  
    print("You didn't enter a number.")  
else:  
    print(f"{x} + {y} = {x+y}")
```

You didn't enter a number.

Exercise 10.7: Addition Calculator.

Wrap your code from Exercise 10-6 in a while loop so the user can continue entering numbers even if they make a mistake and enter text instead of a number.

```
In [ ]: while True:  
    try:  
        x = int(input("Enter a number: "))  
        x = int(x)  
  
        y = input("Enter another number: ")  
        y = int(y)  
    except ValueError:  
        print("You didn't enter a number.")  
    else:  
        print(f"{x} + {y} = {x+y}")  
  
    choice=input("Would you like to continue? (yes/no) ")  
    if choice=='no':  
        break
```

3 + 5 = 8

8 + 5 = 13

3 + 2 = 5

4 + 5 = 9

You didn't enter a number.

3 + 2 = 5

Exercise 10.8: Cats and Dogs.

Make two files, cats.txt and dogs.txt . Store at least three names of cats in the first file and three names of dogs in the second file. Write a program that tries to read these files and print the contents of the file to the

screen. Wrap your code in a `try-except` block to catch the `FileNotFoundException`, and print a friendly message if a file is missing. Move one of the files to a different location on your system, and make sure the code in the `except` block executes properly.

Output with both files in the same directory:

```
In [ ]: fileNames=['cats.txt', 'dogs.txt']

for fileName in fileNames:
    print(f"\nReading file {fileName}:\n")
    try:
        with open(fileName) as f:
            lines=f.read()
            print(lines)
    except FileNotFoundError:
        print(f"Sorry, the file {fileName} does not exist.")
```

Reading file cats.txt:

```
luna
kiki
angel
ori
```

Reading file dogs.txt:

```
hachiko
julius
ceasear
maya
```

Output with one file moved to a different directory:

```
In [ ]: fileNames = ['cats.txt', 'dogs.txt']

for fileName in fileNames:
    print(f"\nReading file {fileName}:\n")
    try:
        with open(fileName) as f:
            lines = f.read()
            print(lines)
    except FileNotFoundError:
        print(f"Sorry, the file {fileName} does not exist.")
```

Reading file cats.txt:

Sorry, the file cats.txt does not exist.

Reading file dogs.txt:

```
hachiko
julius
ceasear
maya
```

Exercise 10.9: Silent Cats and Dogs.

Modify your `except` block in Exercise 10-8 to fail silently if either file is missing.**

Output with both files in the same directory:

```
In [ ]: fileNames = ['cats.txt', 'dogs.txt']

for fileName in fileNames:
    try:
        with open(fileName) as f:
            lines = f.read()
    except FileNotFoundError:
        pass
    else:
        print(f"\nReading file {fileName}:\n")
        print(lines)
```

Reading file cats.txt:

```
luna
kiki
angel
ori
```

Reading file dogs.txt:

```
hachiko
julius
ceasear
maya
```

Output with both files in the same directory:

```
In [ ]: fileNames = ['cats.txt', 'dogs.txt']

for fileName in fileNames:
    try:
        with open(fileName) as f:
            lines = f.read()
    except FileNotFoundError:
        pass
    else:
        print(f"\nReading file {fileName}:\n")
        print(lines)
```

Reading file dogs.txt:

```
hachiko
julius
ceasear
maya
```

Exercise 10.10: Common Words.

Visit Project Gutenberg (<http://gutenberg.org/>) and find a few texts you'd like to analyze. Download the text files for these works, or copy the raw text from your browser into a text file on your computer.

You can use the `count()` method to find out how many times a word or phrase appears in a string. For example, the following code counts the number of times 'row' appears in a string:

```
>>> line = "Row, row, row your boat"
>>> line.count('row')
2
>>> line.lower().count('row')
3
```

Notice how converting the string to lowercase using `Lower()` catches all appearances of the word you're looking for, regardless of how it's formatted.

Write a program that reads the files you found at Project Gutenberg and determines how many times the word 'the' appears in each text. This will be an approximation because it will also count words such as 'then' and 'tehre'. Try counting 'the ', with a space in the string, and see how much you lower your count is.

Counting 'the' with no space:

```
In [ ]: def countCommonWords(filename, word):
    try:
        with open(filename, encoding='utf-8') as f:
            lines=f.read()
    except FileNotFoundError:
        pass
    else:
        wordCount=lines.lower().count(word)
        print(f"The word '{word}' appears {wordCount} times in the file {filename}.")

fileName="Romeo and Juliet.txt"
countCommonWords(fileName, 'the')
```

The word 'the' appears 1571 times in the file Romeo and Juliet.txt.

Counting 'the ' with a space:

```
In [ ]: def countCommonWords(filename, word):
    try:
        with open(filename, encoding='utf-8') as f:
            lines = f.read()
    except FileNotFoundError:
        pass
    else:
        wordCount = lines.lower().count(word)
        print(
```

```

f"The word '{word}' appears {wordCount} times in the file {filename}.")\n\n
fileName = "Romeo and Juliet.txt"
countCommonWords(fileName, 'the ')

```

The word 'the ' appears 847 times in the file Romeo and Juliet.txt.

Exercise 10.11: Favorite Number.

Write a program that prompts for the user's favorite number. Use json.dump() to store this number in a file. Write a separate program that reads in this value and prints the message, "I know your favorite number! It's ____."

```
In [ ]: import json\n\nnumber=input("Enter your favourite number: ")\\n\nwith open('favourite_number.json', 'w') as f:\\n    json.dump(number, f)\\n    print("Your favorite number has been stored!")\\n\\nprint("Magically retrieving your favourite number from the file:")\\nwith open('favourite_number.json') as f:\\n    number=json.load(f)\\n    print(number)
```

Your favorite number has been stored!
Magically retrieving your favourite number from the file:
16

Exercise 10.12: Favorite Number Remembered.

Combine the two programs from Exercise 10-11 into one file. If the number is already stored, report the favorite number to the user. If not, prompt for the user's favorite number and store it in a file. Run the program twice to see that it works.

Output when the number isn't stored

```
In [ ]: import json\n\ntry:\\n    with open('favourite_number_2.json') as f:\\n        number=json.load(f)\\nexcept FileNotFoundError:\\n    number=input("Enter your favourite number: ")\\n    with open('favourite_number_2.json', 'w') as f:\\n        json.dump(number, f)\\n        print("Your favorite number has been stored!")\\nelse:\\n    print("Magically retrieving your favourite number from the file:")\\n    print(number)
```

Your favorite number has been stored!

Output when the number is stored

```
In [ ]: try:
    with open('favourite_number_2.json') as f:
        number = json.load(f)
except FileNotFoundError:
    number = input("Enter your favourite number: ")
    with open('favourite_number_2.json', 'w') as f:
        json.dump(number, f)
    print("Your favorite number has been stored!")
else:
    print("Magically retrieving your favourite number from the file:")
    print(number)
```

Magically retrieving your favourite number from the file:

16

Exercise 10.13: Verify User.

The final listing for `remember_me.py` assumes either that the user has already entered their username or that the program is running for the first time. We should modify it in case the current user is not the person who last used the program.

Before printing a welcome back message in `greet_user()`, ask the user if this is the correct username. If it's not, call `get_new_username()` to get the correct username.

Output when a new user is stored

```
In [ ]: import json

def getStoredUser():
    fileName='username.json'
    try:
        with open(fileName) as f:
            username=json.load(f)
    except FileNotFoundError:
        return None
    else:
        return username

def getNewUser():
    username=input("Enter your name: ")
    fileName='username.json'
    with open(fileName, 'w') as f:
        json.dump(username, f)
    return username

def greetUser():
    username=getStoredUser()
    if username:
```

```
        print(f"Welcome back {username}!")
else:
    username=getNewUser()
    print(f"We'll remember you when you come back, {username}!")

greetUser()
```

We'll remember you when you come back, Muid!

Output when the user is stored

```
In [ ]: def getStoredUser():
    fileName = 'username.json'
    try:
        with open(fileName) as f:
            username = json.load(f)
    except FileNotFoundError:
        return None
    else:
        return username

def getNewUser():
    username = input("Enter your name: ")
    fileName = 'username.json'
    with open(fileName, 'w') as f:
        json.dump(username, f)
    return username

def greetUser():
    username = getStoredUser()
    if username:
        print(f"Welcome back {username}!")
    else:
        username = getNewUser()
        print(f"We'll remember you when you come back, {username}!")

greetUser()
```

Welcome back Muid!