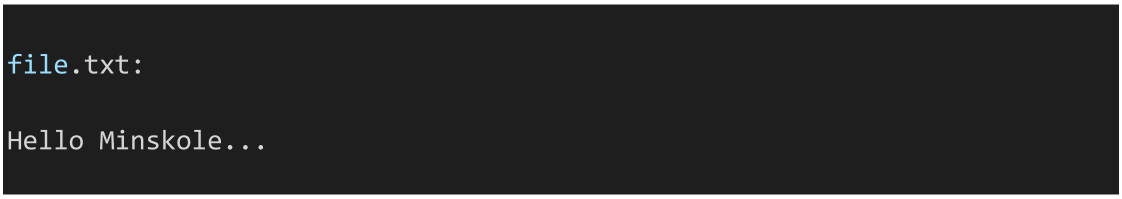
ANSWERS ASSIGNMENT – 11

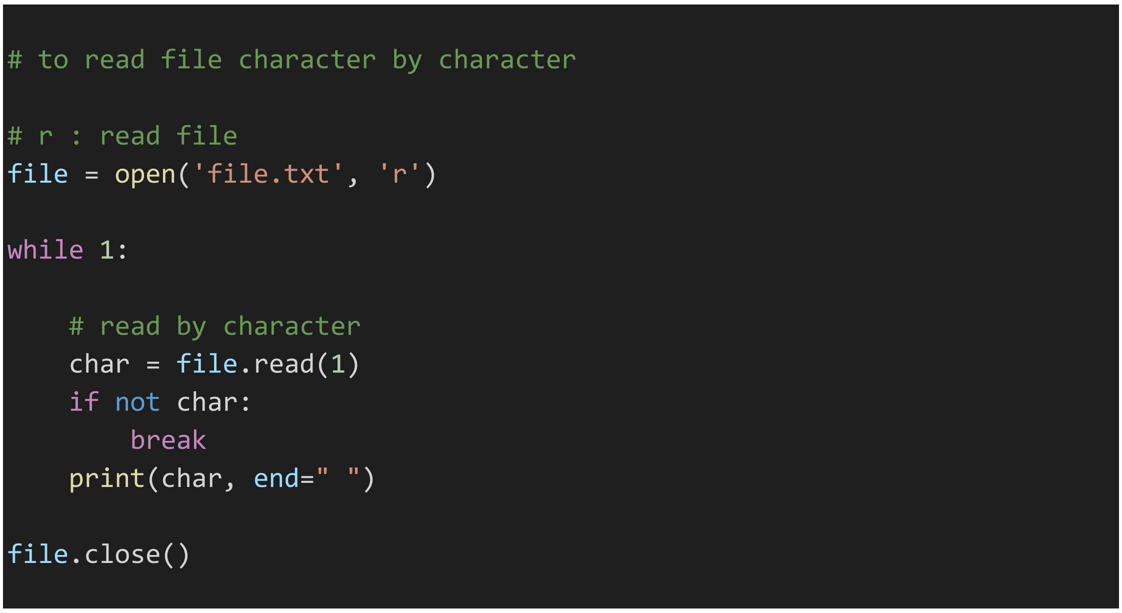
TOPICS – Files in Python

1. **A Python Program to create text file to store individual characters.**
Solution:



```
file.txt:

Hello Minskole...
```

2. **A Python Program to read characters from a text file.**
Solution:



```
# to read file character by character

# r : read file
file = open('file.txt', 'r')

while 1:

    # read by character
    char = file.read(1)
    if not char:
        break
    print(char, end=" ")

file.close()
```

3. A Python Program to store a group of strings into a text file.

Solution:

```
file.txt

Minskole: Knowledge Brings Reform
Email: info@minskole.in
Address: 2nd Floor, Krushna Park, Above Masa Masa Restaurant Yashwant N
agar,
                Kharadi pune, Maharashtra 411014
```

4. A Python Program to read all the strings from the text file and display them.

Solution:

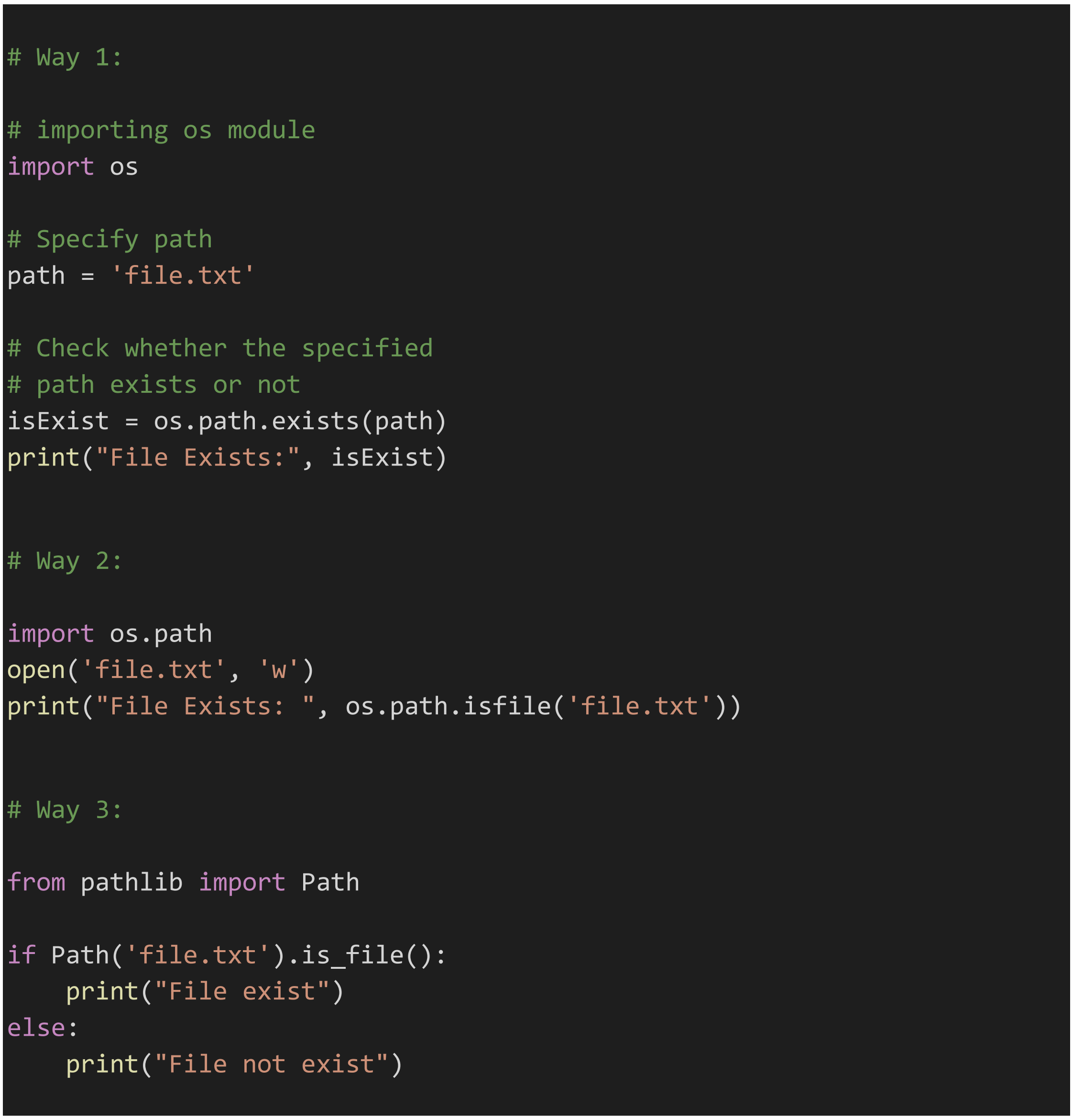
```
# r : read file
file = open('file.txt', 'r')
for i in file:
    print(i, end=" ")
file.close()
```

5. A Python Program to append data to an existing file and then displaying the entire file.

Solution:

```
# w : read file
file1 = open("file.txt", "w")
L = ["Python \n", "Angular \n", "Hadoop"]
file1.writelines(L)
file1.close()

file1 = open("file.txt", "r")
print("Output of Readlines after writing:")
print(file1.read())
print()
file1.close()
```

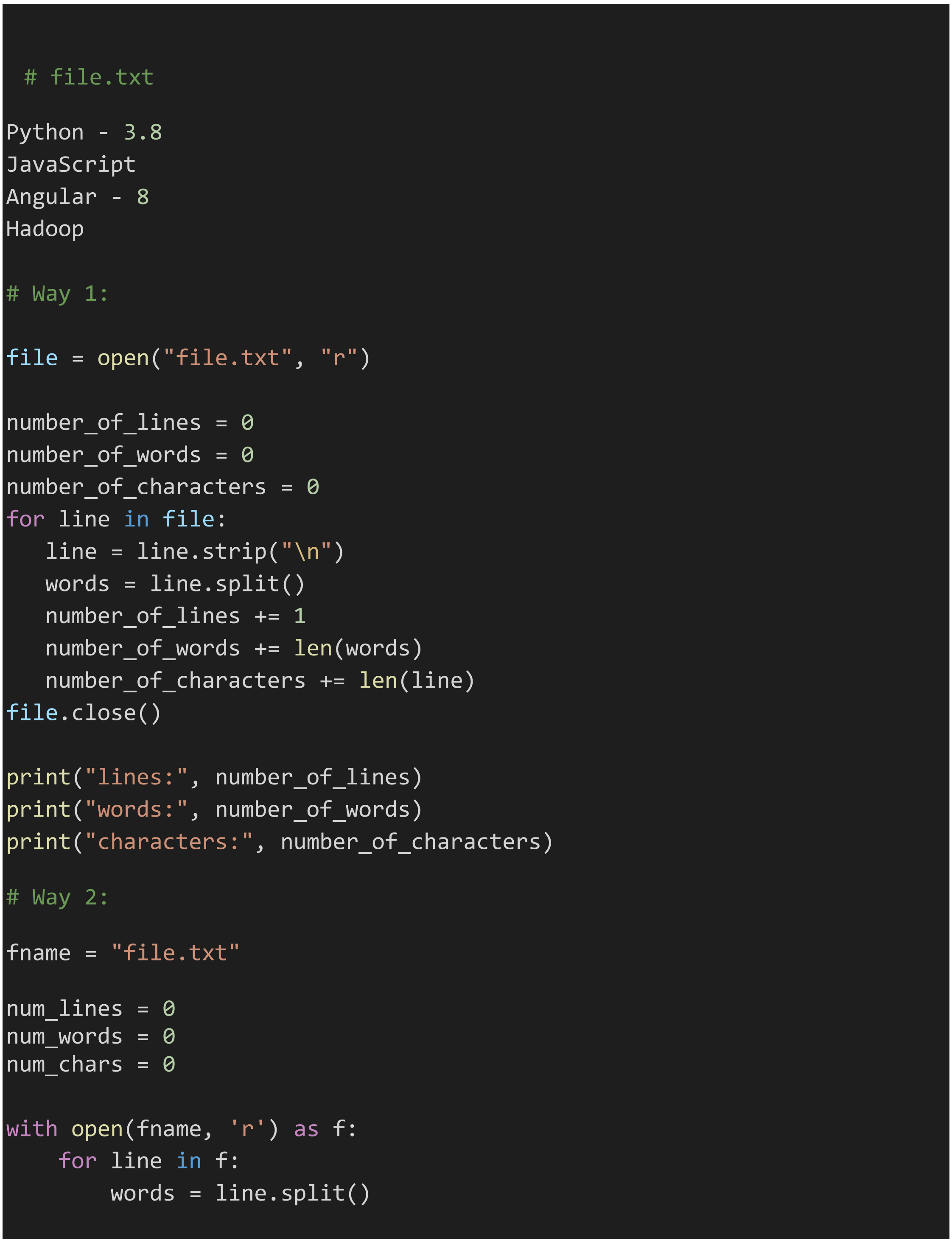
6. A Python Program to know whether a file exists or not.
Solution:



```
# Way 1:

# importing os module
import os

# Specify path
path = 'file.txt'

# Check whether the specified
# path exists or not
isExist = os.path.exists(path)
print("File Exists:", isExist)


# Way 2:

import os.path
open('file.txt', 'w')
print("File Exists: ", os.path.isfile('file.txt'))


# Way 3:

from pathlib import Path

if Path('file.txt').is_file():
    print("File exist")
else:
    print("File not exist")
```

7. A Python Program to count number of lines, words and characters in a text file.

Solution:



```
 # file.txt

Python - 3.8
JavaScript
Angular - 8
Hadoop

# Way 1:

file = open("file.txt", "r")

number_of_lines = 0
number_of_words = 0
number_of_characters = 0
for line in file:
   line = line.strip("\n")
   words = line.split()
   number_of_lines += 1
   number_of_words += len(words)
   number_of_characters += len(line)
file.close()

print("lines:", number_of_lines)
print("words:", number_of_words)
print("characters:", number_of_characters)

# Way 2:

fname = "file.txt"

num_lines = 0
num_words = 0
num_chars = 0

with open(fname, 'r') as f:
    for line in f:
        words = line.split()
        num_lines += 1
        num_words += len(words)
        num_chars += len(line)

f.close()

print("lines:", num_lines)
print("words:", num_words)
print("characters:", num_chars)
```

8. A Python Program to copy an image file into another file.
Solution:

9. A Python Program to use 'with' to open and write some string into a file.
Solution:

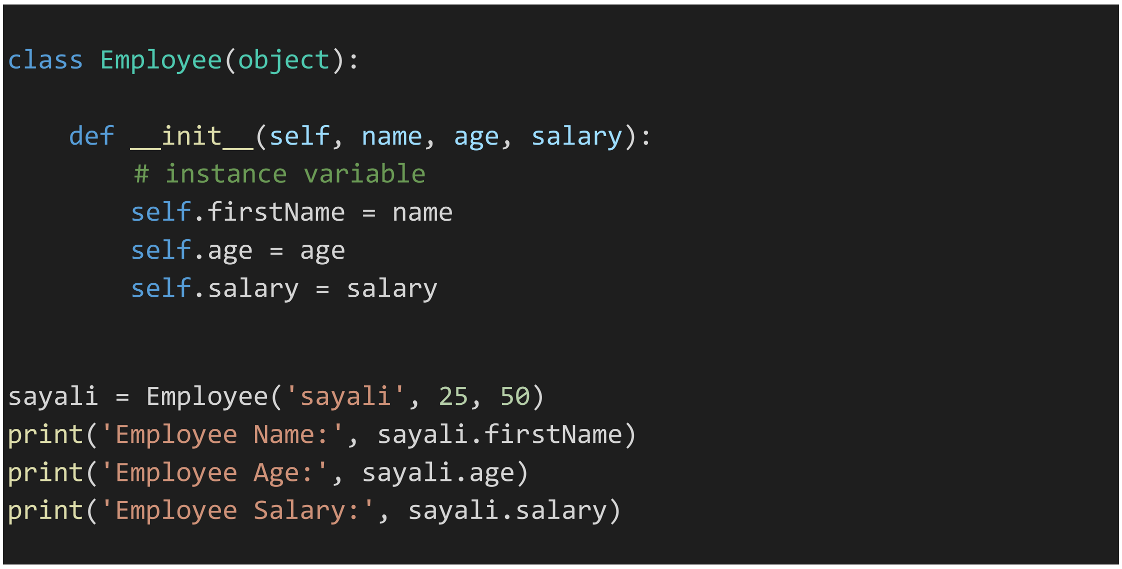
```
with open("file.txt", "w") as file1:
    List = ["Chinmay \n", "Sarika \n", "Pratiksha\n", "Sayali\n",
              "Nikita"]
    file1.writelines(List)
    file1.close()
```

10. A Python Program to use 'with' to open file and read data from it.
Solution:

```
with open("file.txt", "w") as file1:
    List = ["Chinmay \n", "Sarika \n", "Pratiksha\n", "Sayali\n",
             "Nikita"]
    file1.writelines(List)
    file1.close()

with open("file.txt", "r") as file1:
    print("Output of Readlines after writing:")
    print(file1.read())
    print()
    file1.close()
```

11. A Python Program to create Employee class with employee details as instance variables.

Solution:



```
class Employee(object):

    def __init__(self, name, age, salary):
        # instance variable
        self.firstName = name
        self.age = age
        self.salary = salary


sayali = Employee('sayali', 25, 50)
print('Employee Name:', sayali.firstName)
print('Employee Age:', sayali.age)
print('Employee Salary:', sayali.salary)
```

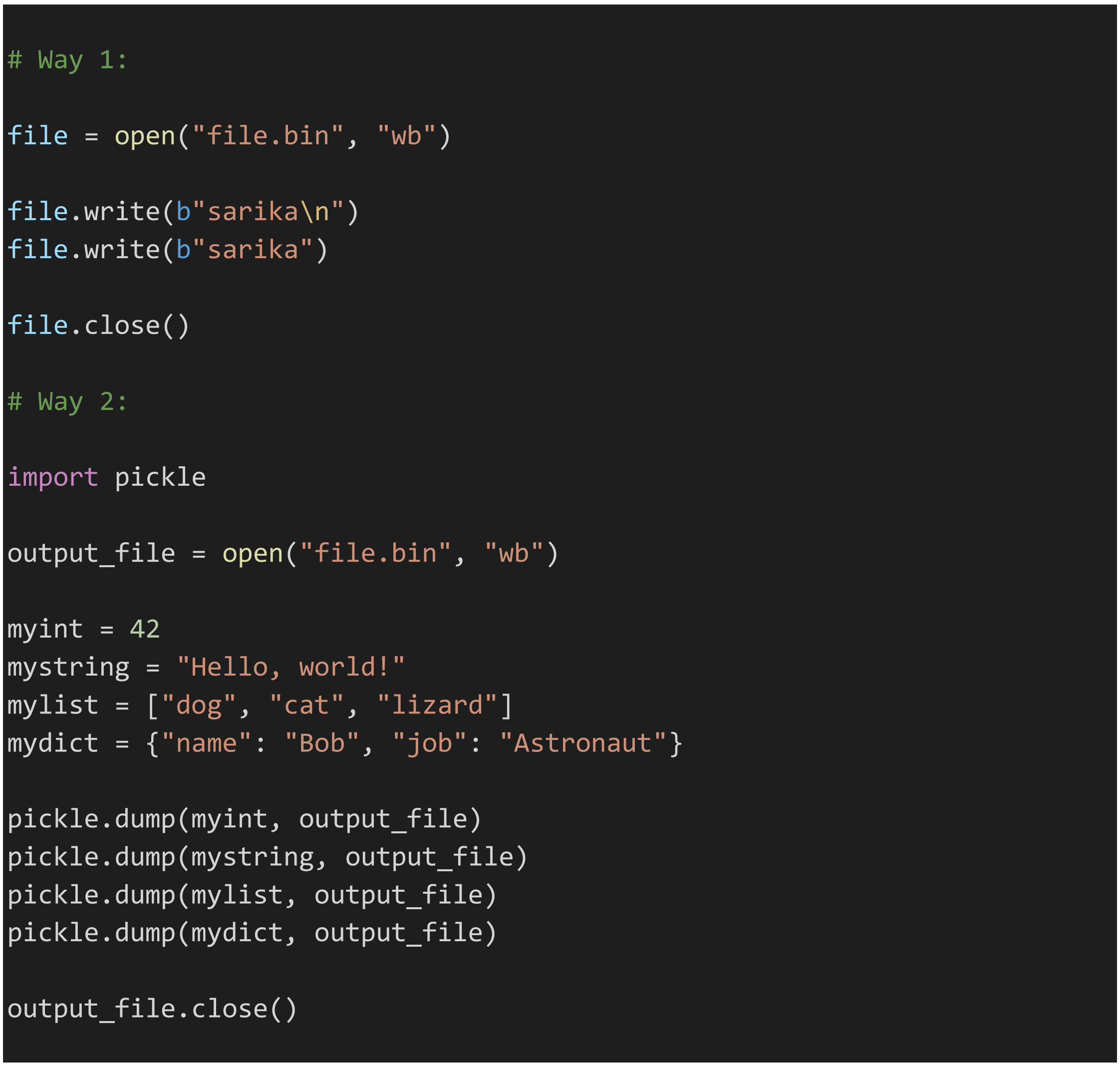
12. A Python Program to pickle Employee class object.

Solution:

13. A Python Program to unpickle Employee class object.

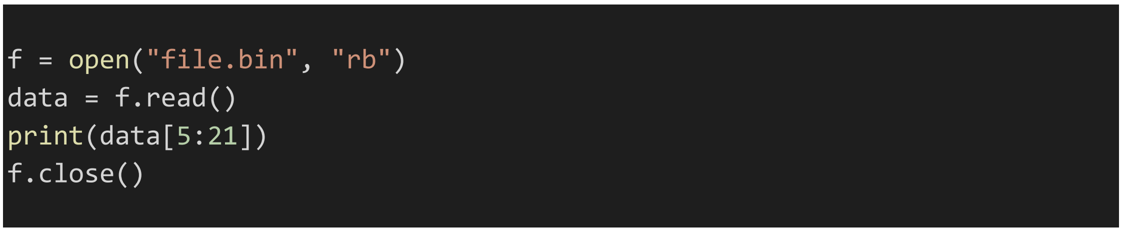
Solution:

14.A Python Program to create a binary file and store a few records.
Solution:



```
# Way 1:

file = open("file.bin", "wb")

file.write(b"sarika\n")
file.write(b"sarika")

file.close()

# Way 2:

import pickle

output_file = open("file.bin", "wb")

myint = 42
mystring = "Hello, world!"
mylist = ["dog", "cat", "lizard"]
mydict = {"name": "Bob", "job": "Astronaut"}

pickle.dump(myint, output_file)
pickle.dump(mystring, output_file)
pickle.dump(mylist, output_file)
pickle.dump(mydict, output_file)

output_file.close()
```

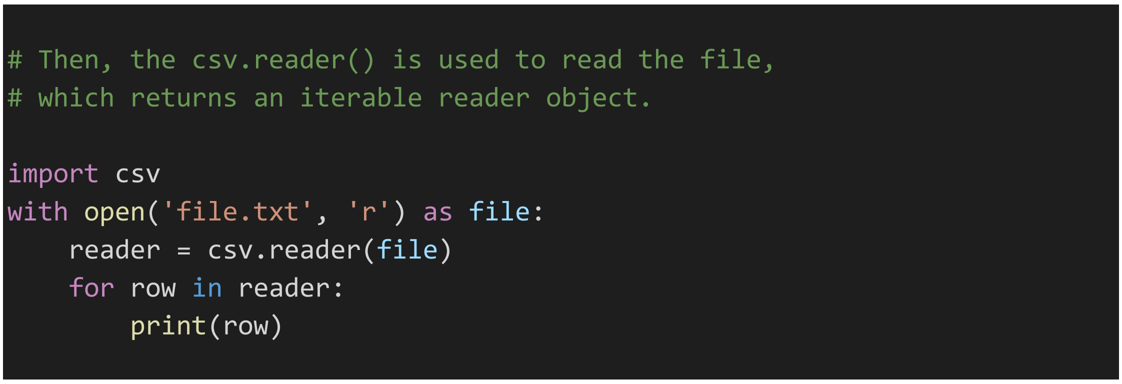
15.A Python Program to randomly access a record from a binary file.
Solution:



```
f = open("file.bin", "rb")
data = f.read()
print(data[5:21])
f.close()
```

16. A Python Program to search for city name in the file and display the record number that contains the city name.

Solution:



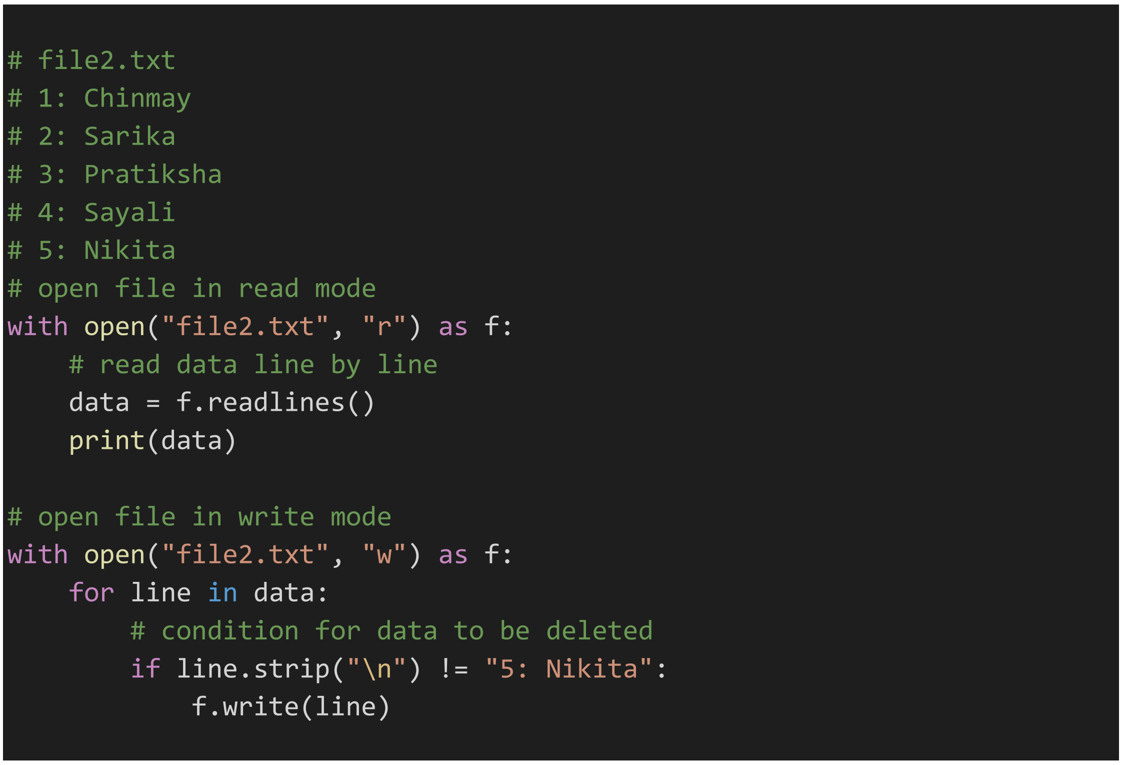
```python
# Then, the csv.reader() is used to read the file,
# which returns an iterable reader object.

import csv
with open('file.txt', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

17. A Python Program to update or modify a record in binary file.

Solution:

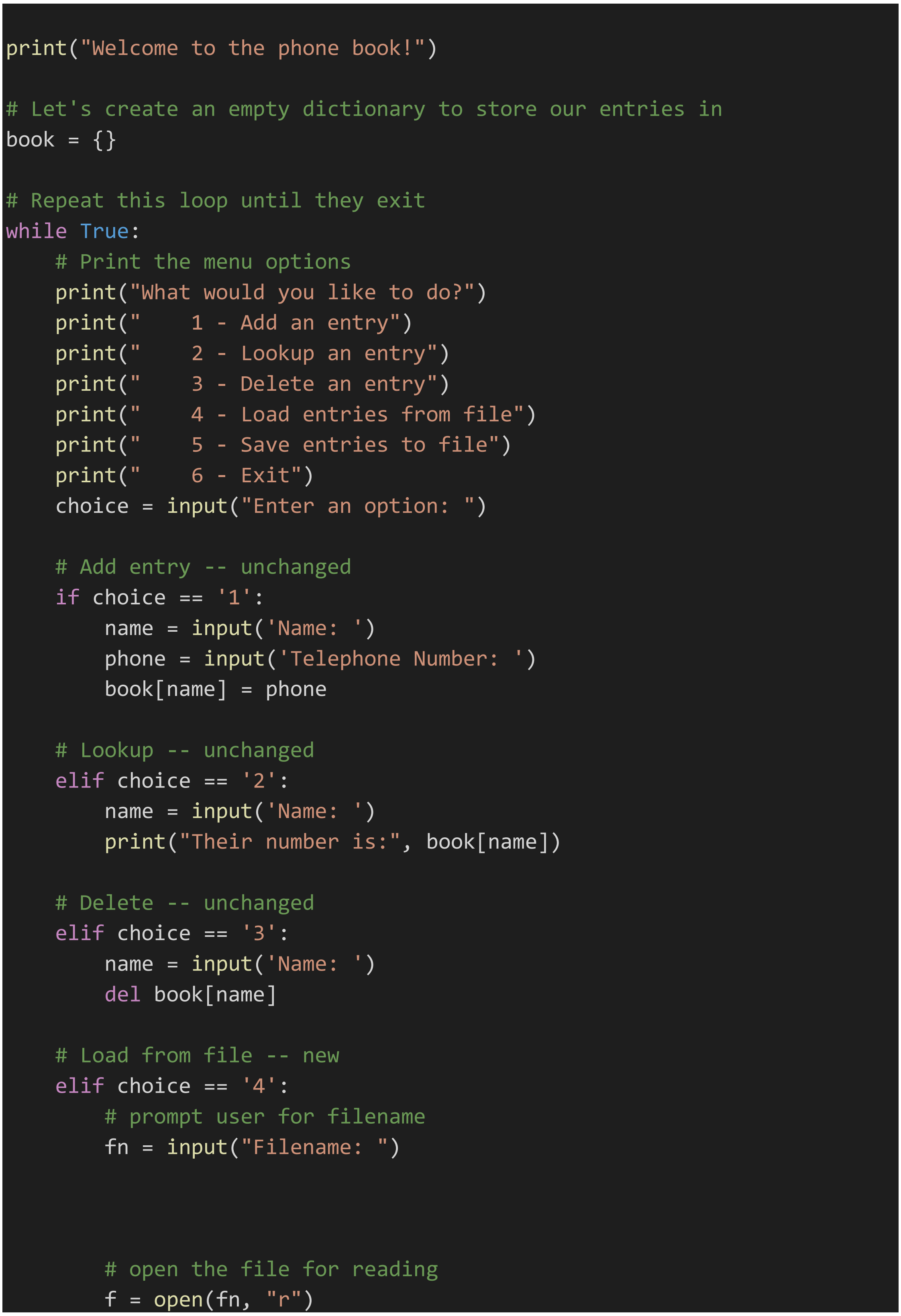
18. A Python Program to delete a specific record from the binary file.

Solution:

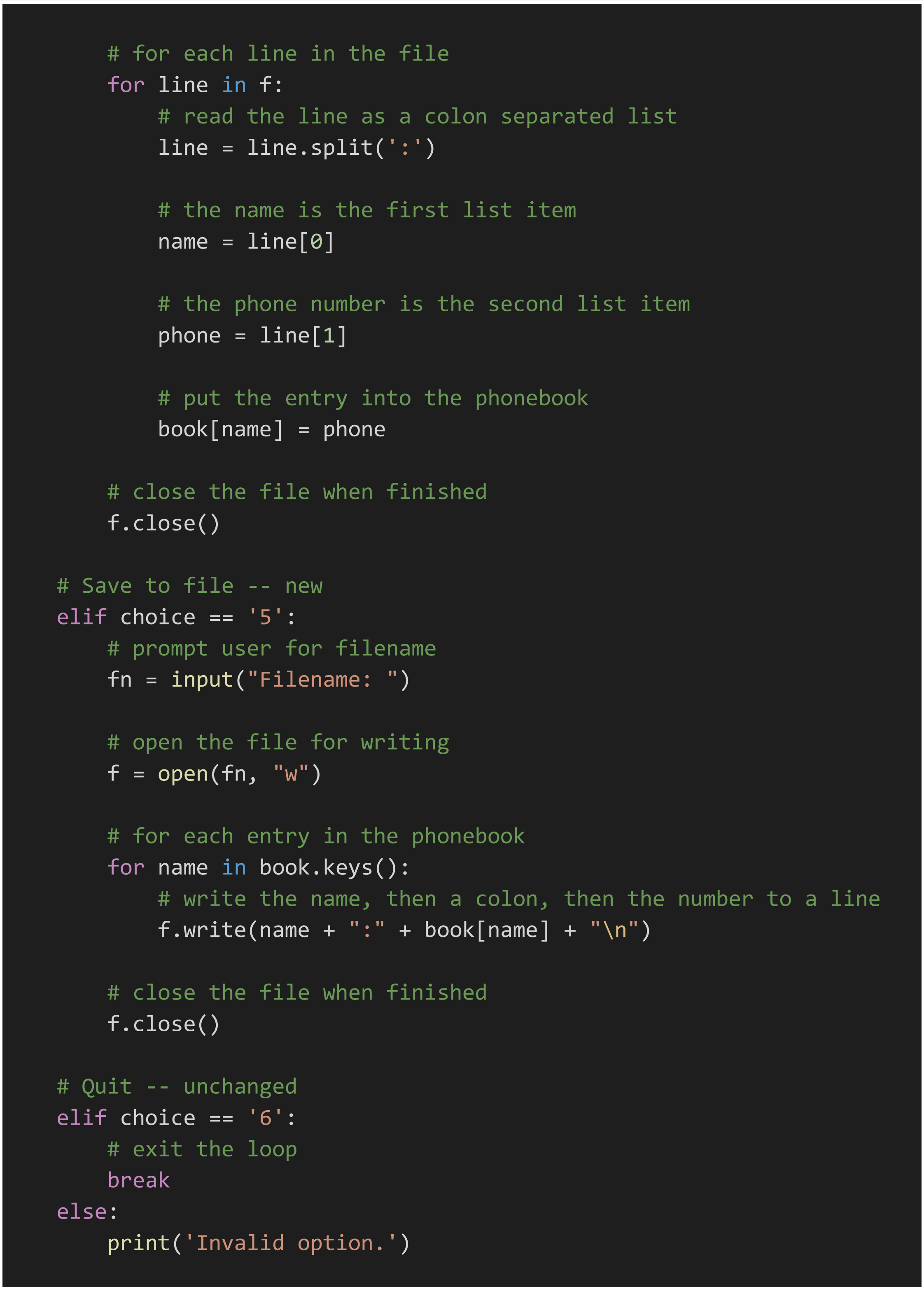


```python
# file2.txt
# 1: Chinmay
# 2: Sarika
# 3: Pratiksha
# 4: Sayali
# 5: Nikita
# open file in read mode
with open("file2.txt", "r") as f:
    # read data line by line
    data = f.readlines()
    print(data)

# open file in write mode
with open("file2.txt", "w") as f:
    for line in data:
        # condition for data to be deleted
        if line.strip("\n") != "5: Nikita":
            f.write(line)
```

19.A Python Program to create a phone book with names and phone number.
Solution:



```python
print("Welcome to the phone book!")

# Let's create an empty dictionary to store our entries in
book = {}

# Repeat this loop until they exit
while True:
    # Print the menu options
    print("What would you like to do?")
    print("    1 - Add an entry")
    print("    2 - Lookup an entry")
    print("    3 - Delete an entry")
    print("    4 - Load entries from file")
    print("    5 - Save entries to file")
    print("    6 - Exit")
    choice = input("Enter an option: ")

    # Add entry -- unchanged
    if choice == '1':
        name = input('Name: ')
        phone = input('Telephone Number: ')
        book[name] = phone

    # Lookup -- unchanged
    elif choice == '2':
        name = input('Name: ')
        print("Their number is:", book[name])

    # Delete -- unchanged
    elif choice == '3':
        name = input('Name: ')
        del book[name]

    # Load from file -- new
    elif choice == '4':
        # prompt user for filename
        fn = input("Filename: ")



        # open the file for reading
        f = open(fn, "r")
```



```
        # for each line in the file
        for line in f:
            # read the line as a colon separated list
            line = line.split(':')

            # the name is the first list item
            name = line[0]

            # the phone number is the second list item
            phone = line[1]

            # put the entry into the phonebook
            book[name] = phone

        # close the file when finished
        f.close()

    # Save to file -- new
    elif choice == '5':
        # prompt user for filename
        fn = input("Filename: ")

        # open the file for writing
        f = open(fn, "w")

        # for each entry in the phonebook
        for name in book.keys():
            # write the name, then a colon, then the number to a line
            f.write(name + ":" + book[name] + "\n")

        # close the file when finished
        f.close()

    # Quit -- unchanged
    elif choice == '6':
        # exit the loop
        break
    else:
        print('Invalid option.')
```

20. A Python Program to use mmap and performing various operations on a binary file.

Solution:

```
import mmap
import contextlib

with open('file.txt', 'r') as f:
    with contextlib.closing(mmap.mmap(f.fileno(), 0, access=mmap.ACCESS
_READ)) as m:
        print('First 10 bytes via read :', m.read(10))
        print('First 11 bytes via slice:', m[:11])
        print('2nd   10 bytes via read :', m.read(10))
```

21. A Python Program to compress the contents file.

Solution:

```
# Way 1:

from zipfile import *
f = ZipFile('E:\\python\\Ass\\Files.Zip', 'w', ZIP_DEFLATED)
f.write('E:\\python\\Ass\\file2.txt')
print('zip file created')

# Way 2:
import zipfile

file_zip = zipfile.ZipFile('E:\\python\\qun\\PythonLogicalQun.zip',
          'w')
file_zip.write('E:\\python\\qun\\PythonLogicalQun.docx', compress_type
                 = zipfile.ZIP_DEFLATED)

file_zip.close()
```

22.A Python Program to unzip of the contents file that are available in a zip file.
Solution:

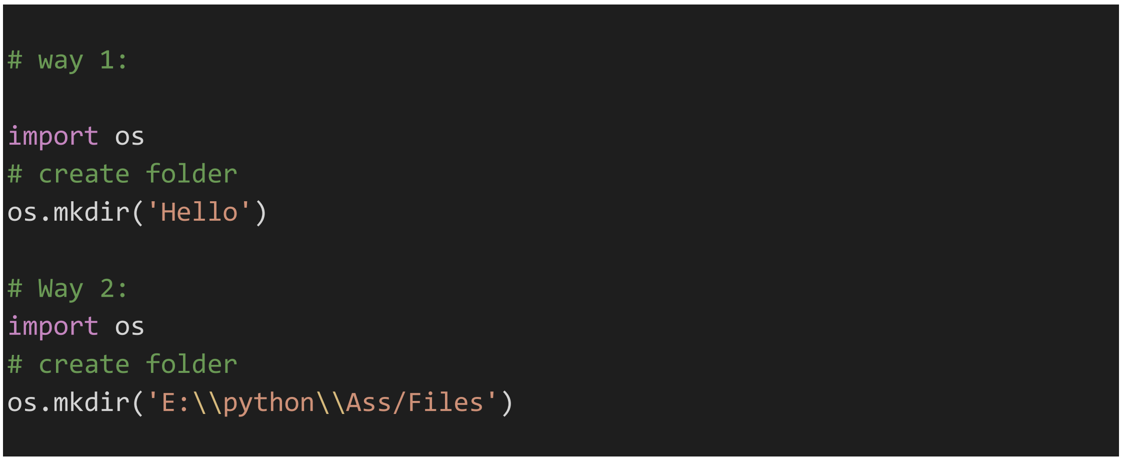
```
import zipfile

file_zip = zipfile.ZipFile('E:\\python\\qun\\PythonLogicalQun.zip')

for file in file_zip.namelist():
    file_zip.extract(file, 'E:\\python\\qun\\PythonLogicalQun')

file_zip.close()
```

23.A Python Program to know the current working directory.
Solution:

```
# way 1:
import os

path = os.getcwd()
print(path)
# E:\python\Ass

# Way 2:

import os
path = os.path.dirname(os.path.realpath(__file__))
print(path)
```

24. A Python Program to create sub directory and then sub-sub directory in the current directory.

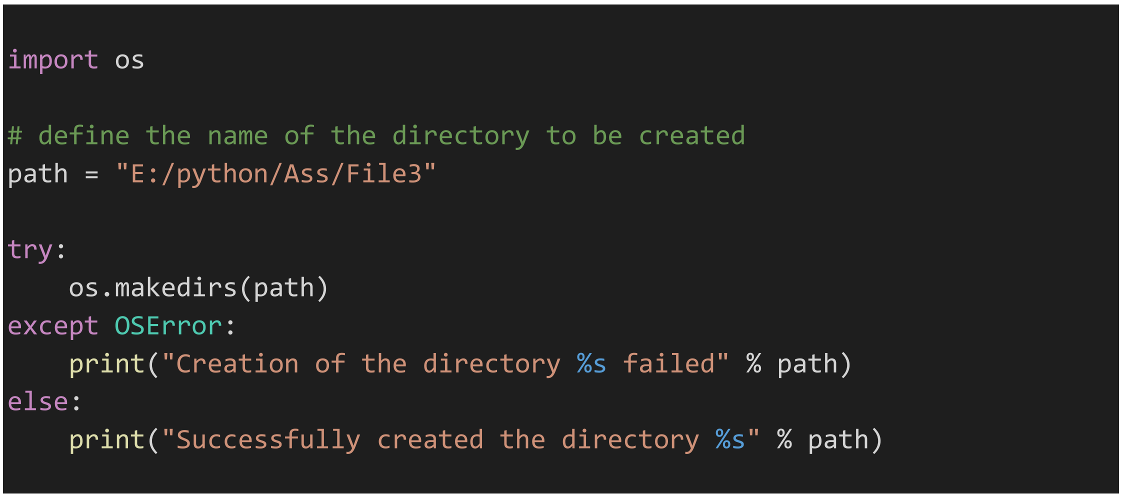
Solution:



```python
# way 1:

import os
# create folder
os.mkdir('Hello')

# Way 2:
import os
# create folder
os.mkdir('E:\\python\\Ass/Files')
```

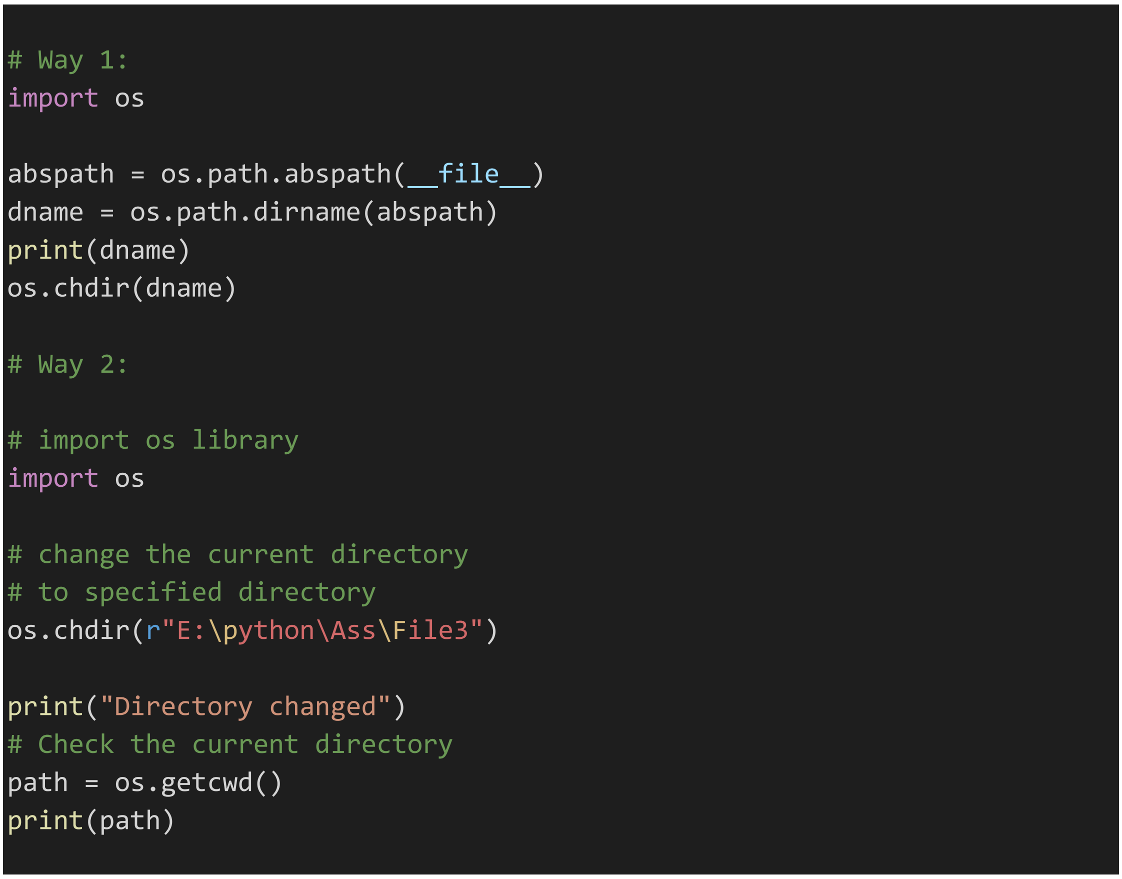
25. A Python Program to use makedirs() function to create sub directory and then sub-sub directory.

Solution:



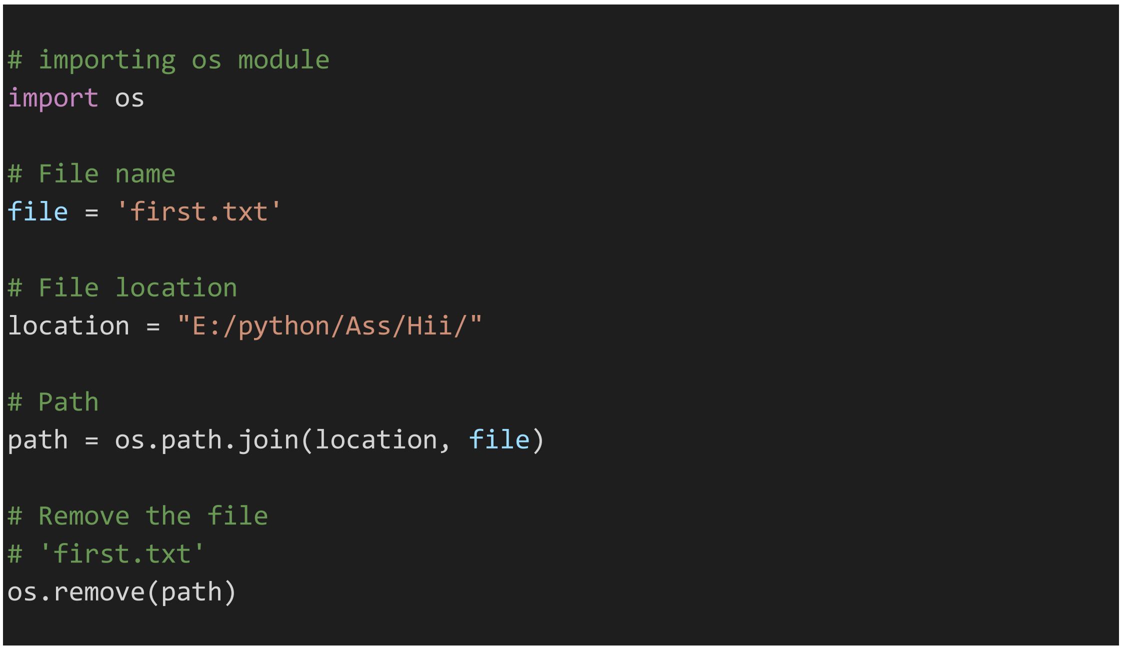
```python
import os

# define the name of the directory to be created
path = "E:/python/Ass/File3"

try:
    os.makedirs(path)
except OSError:
    print("Creation of the directory %s failed" % path)
else:
    print("Successfully created the directory %s" % path)
```

26.A Python Program to change another directory.
Solution:



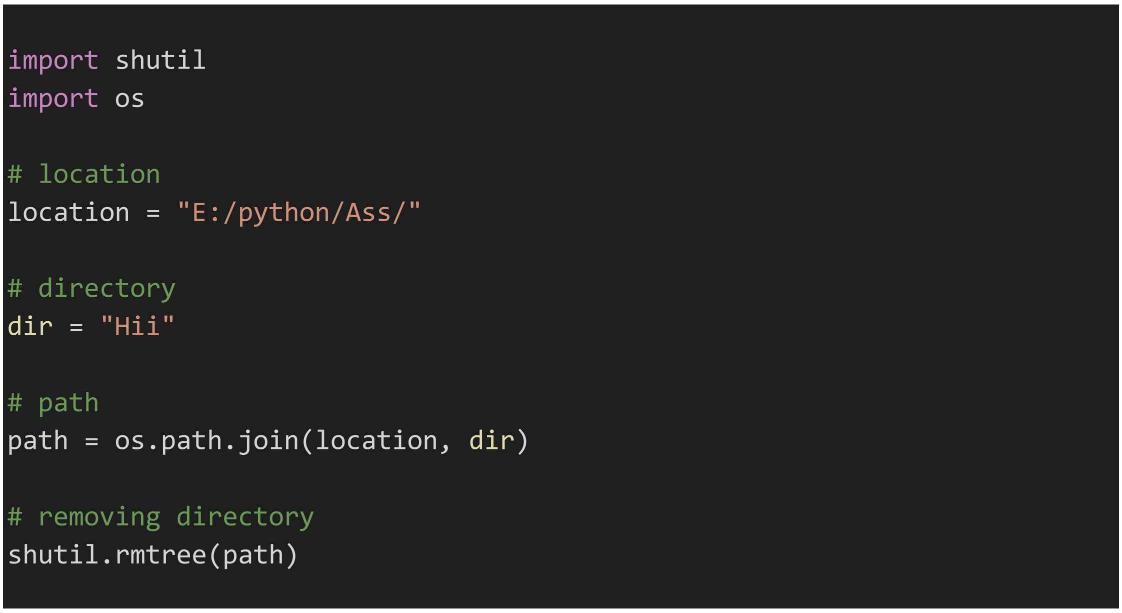
```
# Way 1:
import os

abspath = os.path.abspath(__file__)
dname = os.path.dirname(abspath)
print(dname)
os.chdir(dname)

# Way 2:

# import os library
import os

# change the current directory
# to specified directory
os.chdir(r"E:\python\Ass\File3")

print("Directory changed")
# Check the current directory
path = os.getcwd()
print(path)
```

27.A Python Program to remove a sub directory that is inside another directory.
Solution:



```
# importing os module
import os

# File name
file = 'first.txt'

# File location
location = "E:/python/Ass/Hii/"

# Path
path = os.path.join(location, file)

# Remove the file
# 'first.txt'
os.remove(path)
```

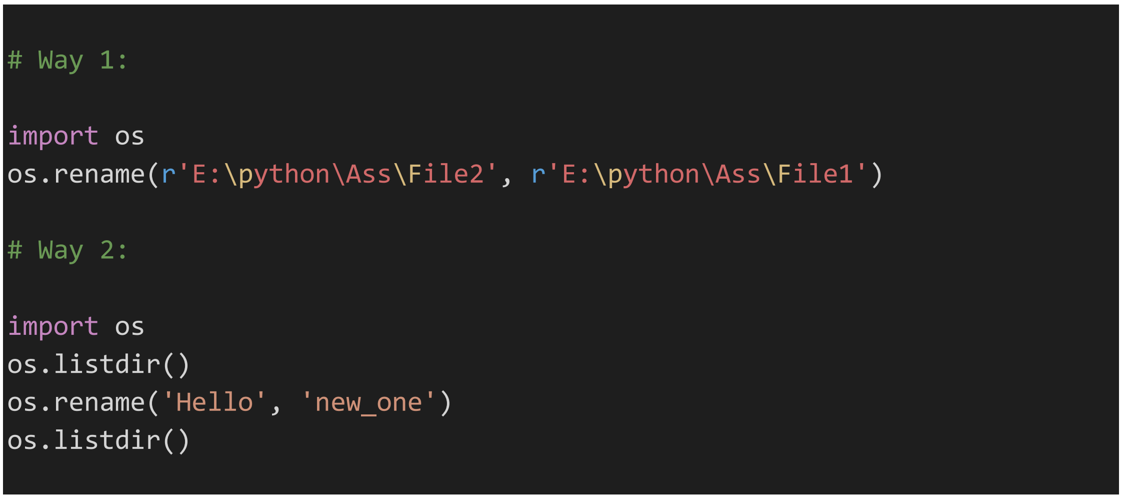
28.A Python Program to remove group of directories in path.
Solution:



```
import shutil
import os

# location
location = "E:/python/Ass/"

# directory
dir = "Hii"

# path
path = os.path.join(location, dir)

# removing directory
shutil.rmtree(path)
```

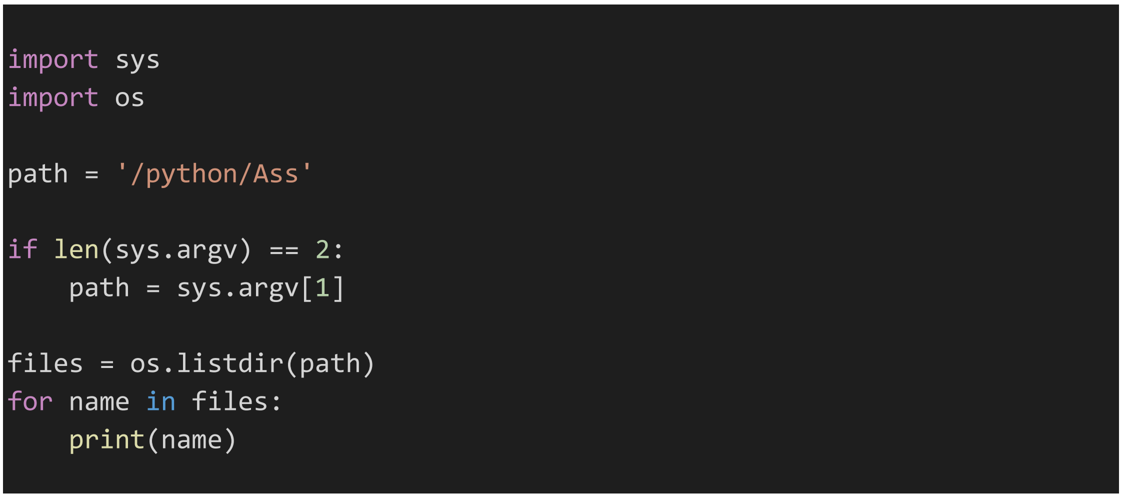
29.A Python Program to rename a directory.

Solution:



```
# Way 1:

import os
os.rename(r'E:\python\Ass\File2', r'E:\python\Ass\File1')

# Way 2:

import os
os.listdir()
os.rename('Hello', 'new_one')
os.listdir()
```

30. A Python Program to display all contents of the current directory.
Solution:



```
import sys
import os

path = '/python/Ass'

if len(sys.argv) == 2:
    path = sys.argv[1]

files = os.listdir(path)
for name in files:
    print(name)
```

31. A Python Program to display Python Program files available in the current directory.
Solution: