Lecture8

October 26, 2024

[16]:

```python
def main():
    outfile = open("philosophers.txt", "w")  # a, w
    outfile.write("John Locke\n")
    outfile.write("David Hume\n")
    outfile.write("Edmund Burke\n")
    outfile.close()


main()
```

[17]:

```python
def main():
    infile = open("philosophers.txt", "r")
    file_contents = infile.read()
    infile.close()  # close the file before printing the contents
    print(file_contents)


main()
```

```
John Locke
David Hume
Edmund Burke
```

[18]:

```python
with open("philosophers.txt", "r") as file:
    contents = file.read()
    print(contents)
```

```
John Locke
David Hume
Edmund Burke
```

[19]:

```python
with open("philosophers.txt", "w") as file:
    file.write("Hello, World!\n")
    file.write("This is a new line.\n")
```

[20]:

```python
with open("philosophers.txt", "a") as file:
    file.write("This line is appended.\n")
```

[21]:

```python
with open("philosophers.txt", "r") as file:
    contents = file.read()
    print(contents)
```

```
Hello, World!
This is a new line.
This line is appended.
```

[22]:

```python
with open("philosophers.txt", "r") as file:
    line = file.readline()
    while line:
        print(line.strip())  # strip() removes "\n" from the end of the line
        line = file.readline()
```

```
Hello, World!
This is a new line.
This line is appended.
```

[23]:

```python
with open("philosophers.txt", "r") as file:
    lines = file.readlines()
    for line in lines:
        print(line.strip())
```

```
Hello, World!
This is a new line.
This line is appended.
```

[46]:

```python
num_days = int(input("For how many days do you have sales? "))
with open("sales.txt", "w") as sales_file:
    for count in range(1, num_days + 1):
        sales = float(input(f"Enter the sales for day #{count}: "))
        sales_file.write(str(sales) + "\n")

print("Data written to sales.txt.")
```

```
Data written to sales.txt.
```

[24]:

```python
with open("sales.txt", "r") as sales_file:
    total = 0
    for line in sales_file:
        amount = float(line)
        total += amount
        print(format(amount, ".2f"))

print("Total sales: ", format(total, ".2f"))
```

```
10.00
20.00
30.00
```

```
40.00
50.00
Total sales:  150.00
```

[25]:

```
with open("sales.txt", "r") as sales_file:
    line = sales_file.readline()
    while line != "":
        amount = float(line)
        print(format(amount, ".2f"))
        line = sales_file.readline()
```

```
10.00
20.00
30.00
40.00
50.00
```

[26]:

```
# Exercise 1
with open("employees.txt", "r") as file:
    lines = file.readlines()
    for i in range(0, len(lines), 3):
        print("Name:", lines[i].strip())
        print("ID:", lines[i+1].strip())
        print("Dept:", lines[i+2].strip())
        print()
```

```
Name: Ingrid Virgo
ID: 4587
Dept: Engineering

Name: Julia Rich
ID: 4588
Dept: Research

Name: Greg Young
ID: 4589
Dept: Marketing
```

[ ]:

```
# get the number of employee records to be create.
num_emps = int(input("How many employee records do you want to create? "))
with open("Lecture08/employees.txt", "w") as emp_file:
    for count in range(1, num_emps + 1):
        print(f"Enter deta for employee #", count, sep="")
        name = input("Name: ")
        id_num = input("ID number: ")
        dept = input("Department: ")
        emp_file.write(name + "\n")
        emp_file.write(id_num + "\n")
        emp_file.write(dept + "\n")
        print()

print("Employee records written to employees.txt.")
```

[27]:
```
import struct

record = (1, "John Doe", 20, 3.75)
with open("records.bin", "wb") as file:
    data = struct.pack(
        "i20sif", record[0], record[1].encode(), record[2], record[3])
        # string has to be encoded to bytes
    file.write(data)
```

[5]:
```
import struct

num_records = int(input("How many records do you want to create? "))
with open("records.bin", "wb") as file:
    for _ in range(num_records):
        id_num = int(input("Enter ID: "))
        name = input("Enter Name: ")
        age = int(input("Enter Age: "))
        gpa = float(input("Enter GPA: "))
        data = struct.pack("i20sif", id_num, name.encode(), age, gpa)
        file.write(data)
print(f"{num_records} records have been written to records.bin")
```

```
6 records have been written to records.bin
```

[6]:
```
import struct
with open("records.bin", "rb") as file:
    data = file.read(struct.calcsize("i20sif"))
    record = struct.unpack("i20sif", data)
    record = (record[0], record[1].decode().strip("\x00"), record[2], record[3])
    print(f"ID: {record[0]}, Name: {record[1]}, Age: {record[2]}, GPA:␣
 ↪{record[3]:.2f}")
```

```
ID: 1, Name: Alice, Age: 20, GPA: 3.25
```

[7]:
```
import struct
with open("records.bin", "rb") as file:
    record_size = struct.calcsize("i20sif")
    while True:
        data = file.read(record_size)
        if not data:
            break
        record = struct.unpack("i20sif", data)
        record = (record[0], record[1].decode().strip("\x00"), record[2],␣
 ↪record[3])
        print(f"ID: {record[0]}, Name: {record[1]}, Age: {record[2]}, GPA:␣
 ↪{record[3]:.2f}")
```

```
ID: 1, Name: Alice, Age: 20, GPA: 3.25
ID: 2, Name: Bob, Age: 24, GPA: 2.67
ID: 3, Name: Harry, Age: 30, GPA: 4.00
ID: 4, Name: Eren, Age: 25, GPA: 3.65
ID: 5, Name: Parry, Age: 19, GPA: 4.00
ID: 6, Name: John, Age: 27, GPA: 3.00
```

[25]:

```
import struct

record_format = 'i20sif'
record_size = struct.calcsize(record_format)

with open("records.bin", "rb") as file:
    file.seek(record_size)

    position = file.tell()
    print(f"Position: {position}")

    data = file.read(record_size)
    record = struct.unpack(record_format, data)
    record = (record[0], record[1].decode().strip("\x00"), record[2], record[3])
    print(f"ID: {record[0]}, Name: {record[1]}, Age: {record[2]}, GPA:␣
 ↪{record[3]}")
```

```
Position: 32
ID: 2, Name: Bob, Age: 24, GPA: 2.6700000762939453
```