



File Handling in Python Class 12

NCERT Solutions

File Handling in Python Class 12 NCERT Solutions

1. Differentiate between:

a) text file and binary file

Answer – Data can be stored in two different types of files: “text files and “binary files”. Text files store the data in a human-readable format like ASCII or Unicode. Binary files store data in a machine-readable format containing non-human-readable characters; a special program is used to read this type of file.

b) `readline()` and `readlines()`

Answer – The `readline()` function reads only one line at a time, whereas `readlines()` reads all lines from a file and returns them as a list.

c) `write()` and `writelines()`

Answer – The write() function is used to write a single string to a file. whereas the writelines() function is used to write multiple strings to a file using an iterable, e.g., a list or tuple.

2. Write the use and syntax for the following methods: a)

open()

```
file_object = open(file_name, mode)
```

b) read()

```
file_object.read(size)
```

c) seek()

```
file_object.seek(offset, from_what)
```

d) dump()

```
import pickle
pickle.dump(data_object, file_object)
```

3. Write the file mode that will be used for opening the following files. Also, write the Python statements to open the following files:

- A text file “example.txt” in both read and write mode
- A binary file “bfile.dat” in write mode
- A text file “try.txt” in append and read mode
- A binary file “btry.dat” in read only mode.

Answer –

A text file “example.txt” in both read and write mode

```
file = open("example.txt", "r+")
```

A binary file “bfile.dat” in write mode

```
file = open("bfile.dat", "wb")
```

A text file “try.txt” in append and read mode

```
file = open("try.txt", "a+")
```

A binary file “btry.dat” in read only mode.

```
file = open("btry.dat", "rb")
```

4. Why is it advised to close a file after we are done with the read and write operations? What will happen if we do not close it? Will some error message be flashed?

Answer – As Python ensures that any unwritten or unsaved data is flushed off (written to the file) before the file is closed, the main reason to close a file after read and write operations is to prevent the application from slowing down. So, once we're done with it, it's always a good idea to close the file.

5. What is the difference between the following set of statements (a) and (b):

a) P = open(“practice.txt”, “r”)

P.read(10)

b) with open(“practice.txt”, “r”) as

P: x = P.read()

Answer – File P will read 10 characters in part “a” but won't print anything; in part “b,” however, it will read every character in the practise file and save it in the variable “x.”

6. Write a command(s) to write the following lines to the text file named hello.txt. Assume that the file is opened in append mode.

“ Welcome my class”

“It is a fun place”

“You will learn and play”

Answer –

```
f = open("hello.txt", "a")
f.write("Welcome my class\n")
f.write("It is a fun place\n")
f.write("You will learn and play\n")
f.close()
```

7. Write a Python program to open the file hello.txt used in question no 6 in read mode to display its contents. What will be the difference if the file was opened in write mode instead of append mode?

Answer –

```
f = open("hello.txt", "r")
x = f.read()
print(x)
f.close()
```

8. Write a program to accept string/sentences from the user till the user enters “END” to. Save the data in a text file and then display only those sentences which begin with an uppercase alphabet.

Answer –

```
f = open("example.txt", "w")
while True:
    sen = input("Enter something ( Enter END for quit ) :- ")
    if sen == "END":
        break
    else:
        f.write(sen + "\n")

f.close()

print("\nLines started with Capital letters:-\n")
f = open("example.txt", "r")
```

```

data = f.readlines()

for i in data:
    if i.strip() and i[0].isupper():
        print(i.strip())

f.close()

```

9. Define pickling in Python. Explain serialization and deserialization of Python object.

Answer – When serialising and deserializing a Python object structure, pickle is primarily employed. It involves transforming a Python object into a byte stream in order to store it in a file or database, keep programme state consistent across sessions, or send data over the network.

10. Write a program to enter the following records in a binary file:

- Item No integer
 - Item_Name string
 - Qty integer
 - Price float
 - Number of records to be entered should be accepted from the user.
- Read the file to display the records in the following format:
- Item No:
 - Item Name :
 - Quantity:
 - Price per item:
 - Amount: (to be calculated as Price * Qty

Answer –

```

import pickle

file = open("Pathwalla.dat", "wb")

num_records = int(input("Enter number of records to be entered: "))

for _ in range(num_records):
    dic = {}
    Item_No = int(input("Enter Item no.: "))
    Item_Name = input("Enter Item name: ")
    Qty = int(input("Enter Quantity: "))
    Price = float(input("Enter price: "))

```

```
dic["Item_No"] = Item_No
dic["Item_Name"] = Item_Name
dic["Qty"] = Qty
dic["Price"] = Price

pickle.dump(dic, file)
print()

file.close()

print("\nDisplaying records:\n")
file = open("Pathwalla.dat", "rb")

try:
    while True:
        dic = pickle.load(file)
        print(f"Item No.: {dic['Item_No']}")
        print(f"Item Name: {dic['Item_Name']}")
        print(f"Quantity: {dic['Qty']}")
        print(f"Price per item: {dic['Price']}")
        print(f"Amount: {dic['Qty'] * dic['Price']}")
        print()
except EOFError:
    file.close()
```