**ASSIGNMENT 18**

QUESTION 1:

Create a zoo.py file first. Define the hours() function, which prints the string 'Open 9-5 daily'.Then, use the interactive interpreter to import the zoo module and call its hours() function.

Ans:

To complete this task, we would first create a file named zoo.py and define the hours() function inside it. Here's an example of how the zoo.py file would look like:

# zoo.py

def hours():

print('Open 9-5 daily')

Once we have saved the zoo.py file, we can use the interactive interpreter to import the zoo module and call its hours() function. Here's an example of how we can do it:

>>> import zoo

>>> zoo.hours()

Open 9-5 daily

QUESTION 2:

In the interactive interpreter, import the zoo module as menagerie and call its hours() function.

Ans:

To import the zoo module as menagerie and call its hours() function, we can use the following code in the interactive interpreter:

>>> import zoo as menagerie

>>> menagerie.hours()

Open 9-5 daily

In the above code, we use the import statement to import the zoo module and give it the alias menagerie. This allows us to refer to the zoo module using the alias menagerie throughout our code.

Then, we call the hours() function from the menagerie module by prefixing it with the module name (menagerie).

QUESTION 3:

Using the interpreter, explicitly import and call the hours() function from zoo.

Ans:

To explicitly import and call the hours() function from the zoo module in the Python interpreter, we can use the following code:

>>> import zoo as menagerie

>>> menagerie.hours()

Open 9-5 daily

In the above code, we use the import statement to import the zoo module and give it the alias menagerie. This allows us to refer to the zoo module using the alias menagerie throughout our code.

Then, we call the hours() function from the menagerie module by prefixing it with the module name (menagerie).

Open 9-5 daily.

QUESTION 4:

Import the hours() function as info and call it.

Ans:

To import the hours() function from the zoo module and call it using the alias info, we can use the following code in the Python interpreter:

>>> import zoo as menagerie

>>> menagerie.hours()

Open 9-5 daily

In the above code, we use the import statement to import the zoo module and give it the alias menagerie. This allows us to refer to the zoo module using the alias menagerie throughout our code.

Then, we call the hours() function from the menagerie module by prefixing it with the module name (menagerie).

When you run this code in the interactive interpreter, it will output:

Open 9-5 daily

as the result of calling the hours() function from the menagerie module.

#QUESTION 5:

Create a plain dictionary with the key-value pairs ‘a’: 1, ‘b’: 2, and ‘c’: 3, and print it out.

Ans:

plain\_dict = {'a': 1, 'b': 2, 'c': 3}

print(plain\_dict)

In this code, we define a dictionary named plain\_dict with the desired key-value pairs using the curly braces {} and separating each pair with a comma. The keys ('a', 'b', 'c') are mapped to their respective values (1, 2, 3) using the colon : notation.

Then, we use the print() function to print the dictionary plain\_dict.

#QUESTION 6:

Make an OrderedDict called fancy from the same pairs listed in 5 and print it. Did it print in the same order as plain?

Ans:

from collections import OrderedDict

plain\_dict = {'a': 1, 'b': 2, 'c': 3}

fancy = OrderedDict(plain\_dict)

print(fancy)

In this code, we import the OrderedDict class from the collections module. We then create an OrderedDict called fancy by passing the plain\_dict dictionary to the OrderedDict constructor.

Finally, we use the print() function to print the fancy OrderedDict.

OrderedDict([('a', 1), ('b', 2), ('c', 3)])

The OrderedDict preserves the order of the key-value pairs as they were inserted, so it will print the same order as the plain dictionary.

#QUESTION 7:

Make a default dictionary called dict\_of\_lists and pass it the argument list. Make the list

dict\_of\_lists[‘a’] and append the value ‘something for a’ to it in one assignment. Print dict\_of\_lists[‘a’].

Ans:

To create a defaultdict called dict\_of\_lists and add a value to the list associated with the key 'a' in one assignment, you can use the following code:

from collections import OrderedDict

plain\_dict = {'a': 1, 'b': 2, 'c': 3}

fancy = OrderedDict(plain\_dict)

print(fancy)

In this code, we import the OrderedDict class from the collections module. We then create an OrderedDict called fancy by passing the plain\_dict dictionary to the OrderedDict constructor.

Finally, we use the print() function to print the fancy OrderedDict.

OrderedDict([('a', 1), ('b', 2), ('c', 3)])

The OrderedDict preserves the order of the key-value pairs as they were inserted, so it will print the same order as the plain dictionary.