**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-First, create a zoo.py file with the following content:

# zoo.py

def hours():

print('Open 9-5 daily')

Save the file zoo.py. Then, you can use the interactive interpreter to import the zoo module and call its hours() function:

import zoo

zoo.hours()

##Output: Open 9-5 daily

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

Ans-n the interactive interpreter, you can import the zoo module as menagerie and call its hours() function:

import zoo as menagerie

menagerie.hours()

Output: Open 9-5 daily

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

Ans-Using the interpreter, you can explicitly import and call the hours() function from zoo:

from zoo import hours

hours()

Output: Open 9-5 daily

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

Ans-You can import the hours() function as info and call it:

from zoo import hours as info

info()

Output: Open 9-5 daily

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

Ans-To create a plain dictionary with the key-value pairs 'a': 1, 'b': 2, and 'c': 3, you can use the following code and print it:

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

print(plain\_dict)

Output: {'a': 1, 'b': 2, 'c': 3}

**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-To create an OrderedDict called fancy from the same key-value pairs and print it, you can use the following code:

from collections import OrderedDict

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

print(fancy) ##output: OrderedDict([('a', 1), ('b', 2), ('c', 3)])

Yes, the OrderedDict fancy will print in the same order as the plain dictionary, unlike regular dictionaries where the order may vary.

**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 make a defaultdict called dict\_of\_lists and append the value 'something for a' to dict\_of\_lists['a'] in one assignment, you can use the following code and print dict\_of\_lists['a']:

from collections import defaultdict

dict\_of\_lists = defaultdict(list)

dict\_of\_lists['a'].append('something for a')

print(dict\_of\_lists['a'])

##Output: ['something for a']

The value 'something for a' is appended to dict\_of\_lists['a'] in one assignment, and then it is printed.