

Assisted Practice 12.5: Abstraction

Problem Scenario: Write a program to demonstrate abstraction using classes, objects, and methods.

Objective: In this demonstration, we will learn how to perform abstraction.

Expected Output:

Ingredients: tomato onion cottage cheese

Taste: Good

Ingredients: chicken meat beef

Taste: Good too

Steps to Perform:

Step 1: Log in to your LMS account.

Step 2: Open the course “**Big data Hadoop and Spark developer**”.

(Note: The course name reflects depending on the program purchased)

Step 3: On the left side click on the “**PRACTICE LABS**” tab and click on the “**LAUNCH LAB**” button.

Big Data Hadoop and Spark Developer
52% Self-Learning Videos Watched | 0/3 Projects Done

BDH

IMP: Dear learner,
Please note: This lab is configured based on the curriculum covered during the live virtual classes.
All details pertaining to the exercises in this lab are provided in the e-books available in your LMS account.

Instructions:

- When you go to the practice Lab page click on the LAUNCH LAB button and it would redirect you to the login credential page.
- To start a Cloudera manager, click on the Auth URL and enter the username and password as given.
- Similarly, for the web console, navigate to the Auth URL, and enter the credentials.
- For FTP, click on the URL and enter the credentials.

Tools:

Cluster Setup

FTP ,Cloudera Infrastructure ,Hadoop 3.x ,HDFS ,YARN ,Kafka ,Flume ,Sqoop ,Pig ,Hive ,Hbase ,Scala ,Spark ,Spark - Core ,Spark SQL ,Spark GraphX ,Spark .ML ,Cloudera Manager

[You can download the Lab Guides from HERE](#)

Your Labs are ready. **LAUNCH LAB**

Step 4: Again, click on the “**LAUNCH LAB**” button.

Your Lab Setup is Ready!

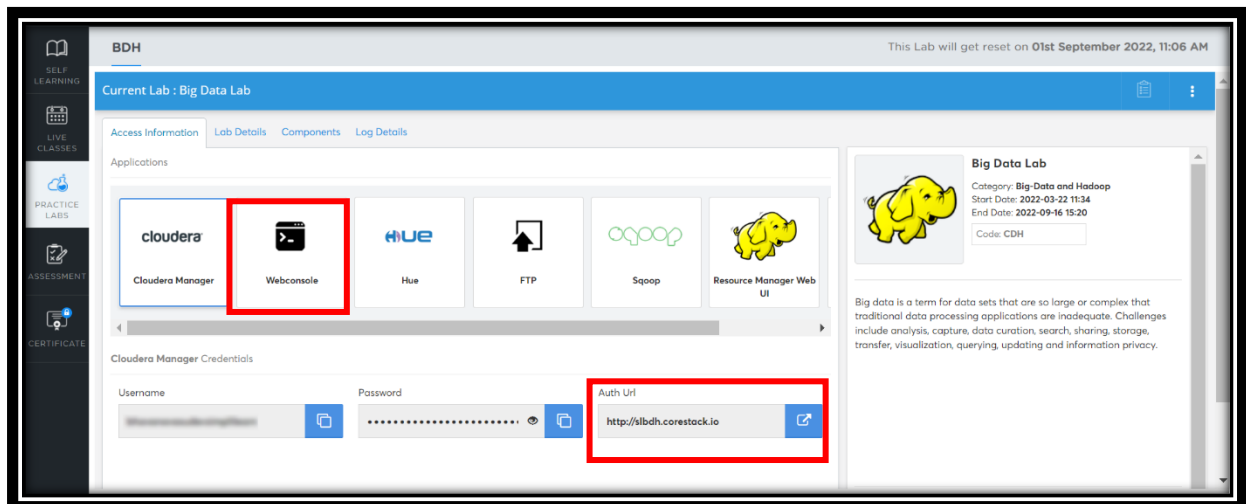
This Lab will get reset on
26th July 2022, 10:02 PM

To ensure continued practice, the reset date is automatically set to **120 days from the last lab start date**. You'll lose all the lab progress on reset activity. Take a backup of your code if you are going to be away for more than 120 days.

LAUNCH LAB

Note: Practice labs are disabled on course expiry date by default.

STEP 5: Click on “**Webconsole**” and click on the “**Auth Url**”.



Step 6: Copy the **"Username"** and the **"Password"** provided to log in to the Web console.

Step 7: Paste the **"Username"** and the **"Password"** on the console and click on enter.

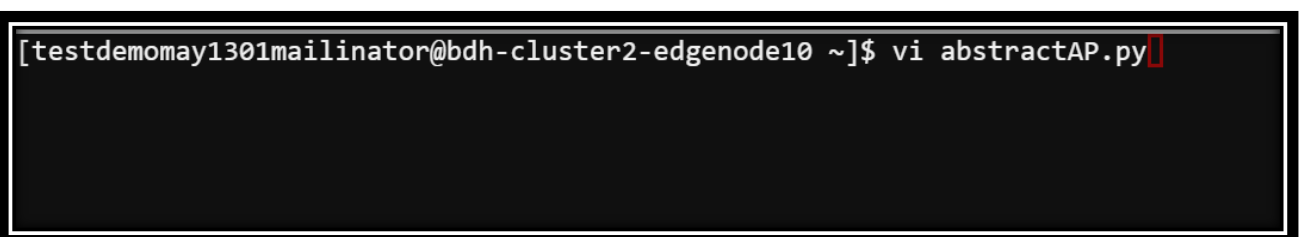
Note: The password will not be visible when pasted on the console



Step 8: Create a python file.

Command:

vi abstractAP.py



The below screen appears:



Step 9: Perform the tasks.

9.1 Import the necessary libraries.

Command:

```
from abc import ABCMeta, abstractmethod
```

9.2 Create a base class with the name Food that contains abstract methods.

Command:

```
class Food():  
  
    __metaclass__ = ABCMeta  
  
    @abstractmethod  
  
    def ingredients(self):  
  
        pass  
  
    def taste(self):  
  
        pass
```

9.3 Create two derived classes.

Command:

```
class Veg(Food):
```

```
def ingredients(self):  
    print("tomato","onion","cottage cheese")  
  
def taste(self):  
    print("Good")  
  
class Nonveg(Food):  
    def ingredients(self):  
        print("chicken","meat","beef")  
  
    def taste(self):  
        print("Good too")
```

9.4 Create objects for the derived class and call the non-abstract methods.

```
obj = Veg()  
  
obj.ingredients()  
  
obj.taste()  
  
  
obj2 = Nonveg()  
  
obj2.ingredients()  
  
obj2.taste()
```

```
from abc import ABCMeta, abstractmethod
class Food():
    __metaclass__ = ABCMeta
    @abstractmethod
    def ingredients(self):
        pass
    def taste(self):
        pass

class Veg(Food):
    def ingredients(self):
        print("tomato","onion","cottage cheese")
    def taste(self):
        print("Good")

class Nonveg(Food):
    def ingredients(self):
        print("chicken","meat","beef")
    def taste(self):
        print("Good too")

obj = Veg()
obj.ingredients()
obj.taste()

obj2 = Nonveg()
obj2.ingredients()
obj2.taste()
```

9.5 Run the code.

Command:

python3 abstractAP.py

```
[testdemomay1301mailinator@bdh-cluster2-edgenode10 ~]$ python3 abstractAP.py
tomato onion cottage cheese
Good
chicken meat beef
Good too
```