Faculty Name: IT

Qualification Name: IT Programming Foundation

Formative Assessment 2

0861 100 395 | ENQUIRY@CTUTRAINING.CO.ZA | CTUTRAINING.AC.ZA

CTU training solutions | Private Tertiary Education

Table of Contents

Formative Assessment 2 Paper

Faculty Name:	Information Technology
Qualification Name:	IT Certificate in Programming Foundation
Module Name:	Programming for Python Semester 1
Module Code:	PRP411
Hand Out:	17 – 04 – 2023
Hand In:	05 – 05 – 2023
Total Marks:	100
Examiner:	Mr. Kobus Mienie
Resources Required:	None

Scenario Question(s)	220 Marks

This Formative Assessment 2 (FA2) contributes 25% towards the final mark.

Instruction(s) to Students

1. Read all instructions carefully before beginning the assessment.
2. Read each question carefully and make sure you understand what is being asked.
3. Show all work and clearly label your answers.
4. When you have finished the assessment, check your work to make sure you have answered all questions to the best of your ability.
5. Submit your work in your zipped folder that should include the source code and a pdf document with screenshots of your code and output. Failure to do so will be penalized.

Scenario Question(s) 100 Marks

Study the scenario and complete the question(s) that follow:

Developing Alarm Clock

You have been employed at CTU Tech, which is a tech company that sells tech equipment. The company has four shops in different areas around the country. All the shops get their stock from the same warehouse. Currently they do not have a system in the warehouse to help them with there sales. You came up with an idea to ease the load, caused because of a lack of a set system. You have received permission, from management, to create an application to be implemented to assist in the sales.

Management has supplied you with a series of requirements to assist you with the creation of the application.

Source: Kobus Mienie (2023).

Question 1

1.1 The Main application and class must be split into two different files named, ctuMain.py and ctuClass.py.

1.2 Create a class named ctuStock with the following attributes: shopName, shopLocation, customers, sales and returns (students can use own attribute naming convention)

1.3 Create four objects from the class (representing the four shops) and assign each with the following defaults: shopName – "Default", shopLocation – "Default", customers - 0, sales - 0 and returns – 0.

1.4 The application must allow for continuous user input (Hint: Use a while loop)

1.5 Create a main menu with the following options: 1. Shop Management, 2. Sales, 3. Returns, 4. Stock, 99. Exit (Refer to the screenshots below)

1.6 When option 1 is selected the following menu must be displayed: 1. Change shop Name, 2. Change shop location, 3. Display current shops, 4. Display all shops information, 0. Back (Refer to the screenshots below)

1.7 When the user selects option 1, all the current towns have to be displayed and the user has to select which town's name to change, the user is then prompted to type in the new shop name (refer to screenshots below)

1.8 Option 2, a list is displayed with the shop name as well as the location, the user has to decide which location they want to change and will be prompted for the new location name. (refer to screenshots below)

1.9 Option 3 will only display all the shop names as well as the location. (refer to screenshots below)

1.10 Option 4 will display all the information for each shop (refer to screenshots)

1.11 Option 2 (Sales) from the main menu will display all the items available to be sold, the user must make a selection and will be prompted for the number of items bought and to select a shop to buy from (refer to screenshots below)

1.12 When an item is sold, the selected shop's customer count has to increase by 1, the sales count has to increase by the amount of items sold and the amount of stock on hand has to be decreased by the amount of items bought.

1.13 Option 3 (returns) from the main menu will prompt the user for an item to be returned as well as the quantity they wish to return and the shop to return the item to. (refer to the screenshots below)

1.14 On a return of an item, the object's (shop's) returns must be increased and sales decreased by the number of items returned

1.15 Option 4 from the main menu will display the following menu: 1. Display stock, 2. Add Stock, 0. Back.

1.16 Option 1 from this menu will display the item, price as well as the quantity on hand for the stock item.

1.17 Option 2 will prompt the user for and Item to add as well as the price of the item and the quantity for the stock on hand.

1.18 Option 99 from the main menu will close the application.

Additional information:

- Functions must be used to display some of the menus
- A minimum of 7 functions have to be used in the application
- Public or private class must be used (private class is prefer but marks will NOT be deducted for using a public class)
- After each "procedure" (name change, sales, etc) a confirmation message has to be displayed to inform the user if the "procedure" was successfully executed
- Stock items, along with the price and quantity, must be kept in a list or dictionary (more than 1 can be used)
- Shop names must be tested (upon change) to make sure that the name is not blank.
- Locations must be tested and is only allowed to be 1 of the following: ["Free State", "Gauteng", "KZN", "Limpopo", "Default"]
- Both tests must be done inside from inside the class (after data has been passed to the class)
- Marks will be allocated for proper/descriptive variable and function names.
- Make precaution for faulty user input (example, if a user types in 5 instead of 1 – 4, tell the user an invalid selection was made)
- Comments are required
- Try to reuse created functions

- Take screenshots of application output, submit screenshots in a pdf document along with rest of the submission)

- All files (.py files, pdf document with screenshot and declaration of authenticity) must be added to a zip file and uploaded to Campus Online

```
Welcome to CTU Technologies

1. Shop Management
2. Sales
3. Returns
4. Stock
99. Exit

Select an option or 99 to exit: 1
Shop Management
1. Change shop Name
2. Change shop location
3. Display current shops
4. Display all shops information
0. Back

Select an option: 1
Change Shop name

Select Shop
1. Default
2. Default
3. Default
4. Default
0. Back
Select an option: 1
Type the new Shop name: Bloemfontein

Shop name was successfully changed to Bloemfontein
```

```
Change Shop Location

Select Shop
1. Bloemfontein, Default
2. Default, Default
3. Default, Default
4. Default, Default
0. Back
Select an option: 1
Enter a location Free State, Gauteng, KZN, Limpopo: Free State

Shop location successfully changed to Free State
```

```
----------------

Shop Name: Bloemfontein
Shop Location: Free State
Number of Customers: 0
Current Sales:0
Returns: 0

----------------

----------------

Shop Name: Default
Shop Location: Default
Number of Customers: 0
Current Sales:0
Returns: 0

----------------

----------------

Shop Name: Default
Shop Location: Default
Number of Customers: 0
Current Sales:0
Returns: 0

----------------

----------------

Shop Name: Default
Shop Location: Default
Number of Customers: 0
Current Sales:0
Returns: 0

----------------
```

```
Current Shops

1. Bloemfontein, Free State
2. Default, Default
3. Default, Default
4. Default, Default
```

[Total = 220 Marks]

Completed Declaration of Authenticity

I_____hereby

(FULL NAME)

declare that the contents of this assignment_____are entirely my work except for the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	Date

Signature:_____