**Contal Services**

**Coding Practical Activity**

**Source code:**

Here is a brief explanation of each method in the CashDispenser class:

**\_\_init\_\_(self, notes: dict)**: This is the constructor method for the class, which is called when an object of the class is created. It initializes the 'notes' attribute of the class with the notes passed as an argument. The 'notes' attribute is a dictionary that keeps track of the number of each type of note available in the cash dispenser.

**get\_notes(self) -> dict**: This method returns the current count of each type of note available in the cash dispenser.

**add\_cash(self, notes: dict)**: This method adds a certain number of each type of note to the cash dispenser. The notes to be added are passed as an argument in the form of a dictionary.

**remove\_cash(self, notes: dict)**: This method removes a certain number of each type of note from the cash dispenser. The notes to be removed are passed as an argument in the form of a dictionary.

**cash\_dispense(self, amount: int):** This method is used to dispense cash for a given amount. It takes the amount to be dispensed as an argument and returns a dictionary of denominations used to dispense that amount. If the amount can't be dispensed due to unavailability of notes it will return False.

**Source code can be found in CashDispenser.py file**

**Test Suite:**

The test suite would contain a set of test cases that would verify that the cash dispenser class is working as expected. Each test case would test a specific functionality of the class, such as initializing the class with a certain set of notes, dispensing a certain amount of cash, and checking that the correct number of notes were dispensed. The test suite would use a testing framework such as unittest or pytest and would include test methods that run the various test cases. Each test method would perform a set of actions such as creating an instance of the cash dispenser class, calling methods on the class, and making assertions about the class's state and behavior. The test suite would be run to check that all test cases pass, indicating that the class is working correctly.

**Code can be found in TestCashDispenser.py file**

**Libraries used:**

The CashDispenser class doesn't require any specific library, it is written in python's standard library. For the test suite, I used the built-in unittest library which is part of python's standard library.

**Instructions to build the application:**

I used PyCharm tool to build and execute my program. Here are the steps:

1. Open PyCharm and create a new project.
2. Create a new file in the project and name it **CashDispenser.py**.
3. Copy the CashDispenser class code and paste it in **CashDispenser.py** file.
4. Click run button to execute the code.
5. Create a new file in the project and name it **TestCashDispenser.py**.
6. Copy the test suite code and paste it in **TestCashDispenser.py** file.
7. In the PyCharm menu, go to Run > Edit Configurations.
8. In the Edit Configuration window, click the + button to create a new configuration.
9. Select Python Tests > pytest and click OK.
10. In the Target field, select the **TestCashDispenser.py** file.
11. Click the Apply button and then the OK button to close the Edit Configuration window.
12. To run the test suite, click on the Run button or use the shortcut key (Ctrl+Shift+F10)

This will run the tests defined in the **TestCashDispenser.py** file using the pytest test runner and will show the results in the Run window of PyCharm. It will also show the test pass or fail status in the left gutter of the editor for each test function.