CS220 Programming Methodology Project 1

In this project, you will write a two-class program that manages customer bank accounts. This task will provide you with the opportunity to learn Python coding.

Objectives:

1. Gain experience with how to write a simple Object-Oriented program in Python.

Requirements:

You will write two class definitions named BankAccount and BankAccountsManager. Each class is defined in its own file. According to PEP convention (see https://www.python.org/dev/peps/pep-0008/), the file names should be: bankaccountsmanager.py and bankaccount.py.

The BankAccount class models the bank account of a customer.

Attributes:

1. id: The unique identifier for an account holder- a string.
2. name: The customer's name= a string.
3. balance: The current balance of the customer's account- float.

The BankAccount class defines these methods:

1. A constructor that initializes the instance with all three attributes.
2. The has_id method takes a target id as a parameter and returns True if the target id is equal to the user's id, False otherwise.
3. The withdraw method takes an amount as a parameter and subtracts the amount from the customer's balance. It raises an Exception* if the amount is greater than the balance. The exception contains the message: 'No action: Amount greater than available balance.'
4. The deposit method takes an amount as a parameter and adds that amount to the customer's balance.
5. The get_balance method returns the current balance.

*For this project, you may use the Exception class for all exceptions raised. In reality it would be best to create your own Exception descendants.

The BankAccountsManager class manages a list of BankAccount instances. Its attribute is a list of BankAccounts. The class defines these methods:

1. A constructor that initializes an empty list of accounts.
2. The make_deposit method takes a customer id and an amount as a parameter. It adds the amount to the customer's balance if the account is found. It raises an Exception if the account is not found.
3. The make_withdrawl method takes a customer id and an amount as a parameter. It subtracts the amount from the customer's balance if the account is found. It raises an Exception if the account is not found.
4. The get_balance method takes a customer id as a parameter. It returns the customer's balance if the account is found. It raises an Exception if the account is not found.

5. The get_account_report method takes a customer id as a parameter. It returns a formatted string representation of the customer's balance if the account is found. It raises an Exception if the account is not found. The output must be formatted exactly in this manner:

```
ID: 12345
Name: Tara
Balance: 159.00
```

   Where there is a single space after the colons and the balance, a float, is formatted to two decimal places.

   Note that the BankAccountsManager class references the BankAccount class. Since they are defined in different modules, you need to import the BankAccount class with this statement in your BankAccountsManager class:

   from bankaccount import BankAccount

   Test your code with a separate Python module. You do not have to turn in that code, and do not include any test code in the two files you are submitting for this project.

   Submit your files in a packed file to the Moodle assignment.