Midterm Lab Task 6. Constructor Activity

Problem 1.

For this program, you are tasked to define the following:

Class - Money:

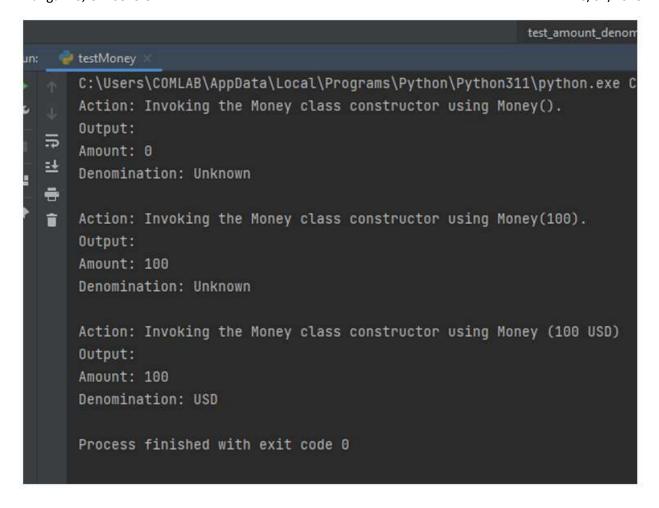
- · Public Properties:
 - amount (type: int): Represents the monetary amount.
 - o denomination (type: str): Specifies the denomination or currency type.
- · Constructor:
 - o __init__(self, amount: int = 0, denomination: str = "Unknown"):
 - . This constructor can be used in three ways:
 - When called with no parameters, it initializes amount to 0 and denomination to "Unknown". This constructor is used when no specific monetary details are provided, setting default values.
 - When called with only the amount as a parameter, it sets the amount
 property accordingly and sets denomination to "Unknown". This
 constructor is useful when only the amount is known, but the
 denomination is not specified.
 - When called with both amount and denomination as parameters, it sets
 the respective properties to these values. This constructor is used when
 complete information about the monetary value, including its
 denomination, is available.

CLASS - MONEY:

TEST CLASS:

```
🎁 main.py
           moneyAccount.py 🔥 testMoney.py
      from moneyAccount import Money
      def test_default():
          money1 = Money()
          print(f"Amount: {money1.amount}")
          print(f"Denomination: {money1.denomination}\n")
      def test_amount():
          money2 = Money(100)
          print(f"Amount: {money2.amount}")
          print(f"Denomination: {money2.denomination}\n")
      def test_amount_denomination():
          money3 = Money( amount 100, denomination: "USD")
          print(f"Amount: {money3.amount}")
          print(f"Denomination: {money3.denomination}")
      if __name__ == "__main__":
          test_default()
          test_amount()
          test_amount_denomination()
```

OUTPUT:



Problem 2.

```
For this program, you are tasked to define the following:
Class - Student:
  · Public Properties:
        o id_number (type: int): A unique identifier for the student.
        o name (type: str): The name of the student.
        o course (type: str): The course the student is enrolled in.

    Methods:

        o __str_() -> str: Returns a string representation of the student's information in
          the format "{id_number} - {name} - {course}".
        o validate_info() -> None: Prints the message "Student information is valid." or
           "Student information is not valid." Indicating whether the student's information is
           valid. Validity criteria include:

    The name should contain only letters.

    The idNumber should be exactly 9 digits long.

Note: Each class should be defined in its own file, with the file name following camelCase
conventions (e.g., bankAccount.py).
```

CLASS - STUDENT:

TEST CLASS:

```
🕏 student.py
                        testStudent.py ×
        from student import Student
         print("Action: Invoking __str__() method with the following Student information:")
            student1 = Student( Id_number: 123456789, name: "John Doe", course: "Computer Science")
        print("ID:", student1.id_number)
print("Name:", student1.name)
print("Course:", student1.course)
print("\n0utput:")
print(student1)
            print(student1)
        student2 = Student( id_number 1234
print("ID:", student2.id_number)
print("Name:", student2.name)
print("Course:", student2.course)
            student2 = Student( id_number: 12345, name: "Jane Doe", course: "Mathematics")
            print("Course:", student2.course)
            print(student2)
            print(*----)
       def test_validate_info(): Tusage
            student3 = Student( id_number: 987654321, name: "Alice123", course: "Physics")
        print("ID:", student3.id_number)
print("Name:", student3.name)
print("Course:", student3.course)
print("\nOutput:")
            student3.validate_info()
            print(*----*)
34 ▷ ' if __name__ == "__main__":
           test_str_method()
            test_str_with_short_id()
            test_validate_info()
```

OUTPUT:

```
C:\Users\Patrick\PycharmProjects\PythonProject2\.venv\Scripts\python.exe C:\Users\
Action: Invoking __str__() method with the following Student information:
ID: 123456789
Name: John Doe
Course: Computer Science
Output:
123456789 - John Doe - Computer Science
Action: Invoking __str__() method with the following Student information:
ID: 12345
Name: Jane Doe
Course: Mathematics
Output:
12345 - Jane Doe - Mathematics
Action: Invoking validate_info() method with the following Student information:
ID: 987654321
Name: Alice123
Course: Physics
Output:
Student information is not valid.
```