Bumanglag, Angeline D.
Mangalino, Chriselle C.

BSCS-C203
10/01/2025

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.
  - `denomination` (type: str): Specifies the denomination or currency type.
- Constructor:
  - `__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:

main.py moneyAccount.py testMoney.py

4 usages

```python
class Money:
    def __init__(self, amount: int = 0, denomination: str = "Unknown"):
        self.amount = amount
        self.denomination = denomination
```

TEST CLASS:

Bumanglag, Angeline D.
Mangalino, Chriselle C.

BSCS-C203
10/01/2025

main.py moneyAccount.py testMoney.py

```python
from moneyAccount import Money

1 usage
def test_default():
    print("Action: Invoking the Money class constructor using Money().")
    money1 = Money()

    print("Output:")
    print(f"Amount: {money1.amount}")
    print(f"Denomination: {money1.denomination}\n")

1 usage
def test_amount():
    print("Action: Invoking the Money class constructor using Money(100).")
    money2 = Money(100)

    print("Output:")
    print(f"Amount: {money2.amount}")
    print(f"Denomination: {money2.denomination}\n")

1 usage
def test_amount_denomination():
    print('Action: Invoking the Money class constructor using Money(100, "USD").')
    money3 = Money( amount: 100,  denomination: "USD")

    print("Output:")
    print(f"Amount: {money3.amount}")
    print(f"Denomination: {money3.denomination}")

if __name__ == "__main__":
    test_default()
    test_amount()
    test_amount_denomination()
```

OUTPUT:

Bumanglag, Angeline D.
Mangalino, Chriselle C.

BSCS-C203
10/01/2025

```
C:\Users\COMLAB\AppData\Local\Programs\Python\Python311\python.exe C
Action: Invoking the Money class constructor using Money().
Output:
Amount: 0
Denomination: Unknown

Action: Invoking the Money class constructor using Money(100).
Output:
Amount: 100
Denomination: Unknown

Action: Invoking the Money class constructor using Money (100 USD)
Output:
Amount: 100
Denomination: USD

Process finished with exit code 0
```

Bumanglag, Angeline D.
Mangalino, Chriselle C.

BSCS-C203
10/01/2025

Problem 2.

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

Class - Student:

- Public Properties:
  - `id_number` (type: int): A unique identifier for the student.
  - `name` (type: str): The name of the student.
  - `course` (type: str): The course the student is enrolled in.
- Methods:
  - `__str__() -> str`: Returns a string representation of the student's information in the format "{id_number} - {name} - {course}".
  - `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:

student.py × testStudent.py

```python
class Student:  4 usages
    def __init__(self, id_number: int, name: str, course: str):
        self.id_number = id_number
        self.name = name
        self.course = course

    def __str__(self) -> str:
        return str(self.id_number) + " - " + self.name + " - " + self.course

    def validate_info(self) -> None:  1 usage
        for ch in self.name:
            if not (ch.isalpha() or ch == " "):
                print("Student information is not valid.")
                return

        if len(str(self.id_number)) != 9:
            print("Student information is not valid.")

        print("Student information is valid.")
```

Bumanglag, Angeline D.
Mangalino, Chriselle C.

BSCS-C203
10/01/2025

TEST CLASS:

student.py testStudent.py

```python
from student import Student

def test_str_method():
    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("\nOutput:")
    print(student1)
    print("-------------------------------------------------------")

def test_str_with_short_id():
    print("Action: Invoking __str__() method with the following Student information:")
    student2 = Student(id_number: 12345, name: "Jane Doe", course: "Mathematics")
    print("ID:", student2.id_number)
    print("Name:", student2.name)
    print("Course:", student2.course)
    print("\nOutput:")
    print(student2)
    print("-------------------------------------------------------")

def test_validate_info():
    print("Action: Invoking validate_info() method with the following Student information:")
    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("-------------------------------------------------------")

# Run the tests
if __name__ == "__main__":
    test_str_method()
    test_str_with_short_id()
    test_validate_info()
```

Bumanglag, Angeline D.
Mangalino, Chriselle C.

BSCS-C203
10/01/2025

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.
-------------------------------------------------------
```