Buccat, Daniel Robert T.
C203

Problem 1:

Code - **File 1**:

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

        # Decide what action message to show based on arguments
        if amount == 0 and denomination == "Unknown":
            self.action = f"Action: Invoking the Money class constructor using
Money()."
        elif denomination == "Unknown":
            self.action = f"Action: Invoking the Money class constructor using
Money({amount})."
        else:
            self.action = f"Action: Invoking the Money class constructor using
Money({amount}, \"{denomination}\")."

    def display(self):
        print(self.action)
        print("Output:")
        print(f"Amount: {self.amount}")
        print(f"Denomination: {self.denomination}")
```

Code - **File 2**:

```
from money import Money

def main():
    print("")
    amount1 = Money()
    amount1.display()

    print("")
    amount2 = Money(100)
    amount2.display()

    print("")
    amount3 = Money(100, "USD")
    amount3.display()

if __name__ == '__main__':
    main()
```

Output:

```
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
```

Problem 2:

Code - **File 1**:

```
class Student:
    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 f"{self.id_number} - {self.name} - {self.course}"

    def validate_info(self) -> None:
        # ID must be exactly 9 digits and name must only have letters + spaces
        if len(str(self.id_number)) == 9 and self.name.replace(" ",
"").isalpha():
            print("Student information is valid.")
        else:
            print("Student information is not valid.")
```

Code - **File 2**:

```
from student import Student

def main():
```

```
    print("Action: Invoking __str__() method with the following Student
information:")
    s1 = Student(123456789, "John Doe", "Computer Science")
    print(f"ID: {s1.id_number}")
    print(f"Name: {s1.name}")
    print(f"Course: {s1.course}")
    print("Output:")
    print(s1)

    print("\nAction: Invoking __str__() method with the following Student
information:")
    s2 = Student(12345, "Jane Doe", "Mathematics")
    print(f"ID: {s2.id_number}")
    print(f"Name: {s2.name}")
    print(f"Course: {s2.course}")
    print("Output:")
    print(s2)

    print("\nAction: Invoking validate_info() method with the following Student
information:")
    s3 = Student(987654321, "Alice123", "Physics")
    print(f"ID: {s3.id_number}")
    print(f"Name: {s3.name}")
    print(f"Course: {s3.course}")
    print("Output:")
    s3.validate_info()

if __name__ == "__main__":
    main()
```

Output:

```
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.

Process finished with exit code 0
```