Part 2:

Explain Code run:

1. The program starts executing from the main method in the **AntiqueShop** class.
2. Inside the main method, a variable item of type Item is declared and initialized to null.
3. A choice variable is declared to store the user's input choice.
4. A Scanner object is created to read input from the user.
5. The program enters a do-while loop to display the menu and perform operations based on the user's choice.
6. Based on the user's choice, the program creates an object of the respective class **(Vase, Statue, or Painting)** and assigns it to the item variable.
7. The program uses casting to call the specific input method **(inputVase(), inputStatue(),** or **inputPainting())** on the item object.
8. Similarly, based on the user's choice, the program uses casting to call the specific output method **(outputVase(), outputStatue(), or outputPainting()).**
9. The program continues to loop until the user enters a choice greater than 4.
10. If the user chooses option 4 (Display the Item), the program checks if item is not null and then uses casting to call the appropriate output method based on the object's actual type.
11. If the user chooses option 4 without creating an object first, the program displays a message indicating that an object needs to be created.
12. The program continues looping until the user enters a choice greater than 4.
13. Once the loop is exited, the program terminates.

Some question:

-What is stored in the static heap, stack, dynamic heap?

+ Static Heap: This is where the class definitions and static variables are stored.

+ Stack: This is where local variables and method calls are stored. The main method, choice variable, and Scanner object are stored in the stack.

+ Dynamic Heap: This is where objects are stored. The item variable is stored in the dynamic heap, and it can reference objects of type Vase, Statue, or Painting.

- What are objects in the program?

+The objects in the program are Vase, Statue, and Painting. These objects are created based on the user's choice and stored in the item variable, which is of type Item.

- What is the item variable storing?

+The item variable is storing a reference to an object of type Item or its subclasses (Vase, Statue, or Painting).

- Why must you cast to call the method inputVase()/outputVase()?

+ Casting is required to call the specific methods (inputVase(), inputStatue(), inputPainting(), outputVase(), outputStatue(), or outputPainting()) because the item variable is declared as type Item. Casting allows the program to treat the item object as an object of its actual type (e.g., Vase, Statue, or Painting) and access the specific methods defined in those classes.

- What is the error thrown when you cast it wrong?

+ If you cast the item variable incorrectly (e.g., casting a Vase object to a Statue), a ClassCastException will be thrown at runtime.

- What methods can you call if you don’t cast the item variable?

+If you don't cast the item variable, you can only call the methods defined in the Item class. The specific methods (inputVase(), inputStatue(), inputPainting(), outputVase(), outputStatue(), or outputPainting()) defined in the subclasses will not be accessible without casting.