Design a class named Person with fields for holding a person's name, address, and telephone number. The class should have the following:

- The no-arg constructor that initializes the object with empty strings for name, address, and phone.
- A constructor that initializes the object with a name, address, and a phone number.
- All appropriate mutator and accessor methods for the class's fields.

Next, design a class named Customer, which extends the Person class. The Customer class should have a field for a customer number and a Boolean field indicating whether the customer wishes to be on a mailing list. The class should also have the following:

- The no-arg constructor that initializes the object with empty strings for name, address, phone, and customerNumber. The mailingList field is set to false.
- Another constructor that initializes the object with a name, address, a phone number, a customer number, and mailing list status.
- All appropriate mutator and accessor methods for the class's fields.

Now, design a class named PreferredCustomer, which extends the Customer class. The PreferredCustomer class should have fields for the amount of the customer's purchases and the customer's discount level. The amount of a customer's discount is determined by the amount of the customer's cumulative purchases in the store as follows:

- When a preferred customer spends $500, he or she gets a 5 percent discount on all future purchases.
- When a preferred customer spends $1,000, he or she gets a 6 percent discount on all future purchases.
- When a preferred customer spends $1,500, he or she gets a 7 percent discount on all future purchases.
- When a preferred customer spends $2,000 or more, he or she gets a 10 percent discount on all future purchases.

The class should also have the following:

- The no-arg constructor that initializes the object with empty strings for name, address, phone, and customerNumber. The mailingList field is set to false. The purchases and discountLevel fields are set to 0.0.
- Another constructor that initializes the object with a name, address, a phone number, a customer number, mailing list status, and amount of purchases.
- All appropriate mutator and accessor methods for the class's fields.

Use the following main method in your Test class and to get a screenshot of the output:

```
public static void main(String[] args)
{
  // Create a Customer object for Julie James.
```

```
    Customer myCustomer = new Customer("Julie James", "123 Main Street", "555-1212",
"147-A049", true);

    // Display the object's data.
    System.out.println("Name: " + myCustomer.getName());
    System.out.println("Address: " + myCustomer.getAddress());
    System.out.println("Telephone: " + myCustomer.getPhone());
    System.out.println("Customer Number: " + myCustomer.getCustomerNumber());
    if (myCustomer.getMailingList())
      System.out.println("Mailing List: YES");
    else
      System.out.println("Mailing List: NO");

    // Create a PreferredCustomer object for Julie James.
    // She has $1750.00 in purchases. That entitles her to a 7% discount on future purchases.
    PreferredCustomer myPCustomer = new PreferredCustomer("Julie James", "123 Main
Street", "555-1212", "147-A049", true, 1750.00);

    // Display the object's data.
    System.out.println("Name: " + myPCustomer.getName());
    System.out.println("Address: " + myPCustomer.getAddress());
    System.out.println("Telephone: " + myPCustomer.getPhone());
    System.out.println("Customer Number: " + myPCustomer.getCustomerNumber());
    if (myPCustomer.getMailingList())
      System.out.println("Mailing List: YES");
    else
      System.out.println("Mailing List: NO");
    System.out.println("Purchases: $" + myPCustomer.getPurchases());
    System.out.println("Discount level: " + myPCustomer.getDiscountLevel());
  }
```