Task 1

In order to fulfill the business requirements, I have created two main classes: Accounts and Customer.

Starting with the Customer accounts, I have implemented inheritance to create a child class by defining an abstract class named PublicAccount and StaffAccount. The Customer class primarily consists of the following information: first name, last name, and ID. To enforce data privacy, I have used encapsulation by declaring some of the information as private. I have utilized getter and setter methods to retrieve and modify this information. Polymorphism has been employed to handle different responses based on Customer information. Upon application startup, the account ID and name are displayed. For Staff Accounts, the information is overridden to include the label "Employee Account". Additionally, the Customer class contains a list of Accounts, allowing multiple accounts to be associated with a single customer.

The Account classes adhere to the same principle. I have created an abstract class that serves as the parent class for three child classes. The fields in this class include Balance, account type, and account ID. Similarly to the Customer class, I have implemented encapsulation by using getter and setter methods and designating certain information as private. To provide relevant information about the account, I have created methods to display details such as Balance, fees, overdraft, and account name. Furthermore, methods for depositing and withdrawing amounts from the account have been implemented. Polymorphism is utilized within this class to override certain information. Abstract methods have also been incorporated, along with the use of the override keyword, as some information was deemed unnecessary in the main class.

Both classes include abstraction methods that retrieve all necessary information and present it in an appropriate format. These methods facilitate the display of essential details in a concise manner, ensuring a user-friendly interface for the application.