

java.io package

Sr. No	Assignment Question
1.	<p>Case Study: The management of the Merchants Bank is looking at automation as a means to save time and effort required in their work. In order to achieve this, the management has planned to automate the following customer transactions:</p> <ul style="list-style-type: none"> • Creating a new account • Withdrawing money from an account • Depositing money in an account • Displaying account details <p>The CEO of the company and a team of experts have chosen your company to provide a solution for the same. Consider yourself to be a part of the team that implements the solution for designing the application.</p> <p>Create an application using I/O streams and text files to implement the solution. Note that the text files should not be editable by a simple text editor. The application should consist of the following classes.</p> <ol style="list-style-type: none"> 1. Account.java 2. Bank.java 3. BankTest.java <p>Each class has a specific purpose and functionality. The descriptions of each class are as follows.</p> <p>Account.java The Account class represents an actual bank account. It stores the following details of a bank account.</p> <ul style="list-style-type: none"> • accountNumber • firstName • lastName • accountbalance <p>These are the instance variables of the class. The Account class will be used by the Bank class to create bank accounts.</p> <p>Bank.java The Bank class creates an array of objects of the Account class to store details of all bank accounts read from the text file. The Bank class contains the following variables for performing various operations on the bank accounts.</p> <ul style="list-style-type: none"> • lastAccountNumber: A static integer variable to specify the number assigned to the last account that was created. • Account[] account: An array of objects to store the account details read from the binary text file. • Scanner input: An instance object to read input from the user at runtime. <p>The account array is initialized in the constructor of the Bank class. The Bank class implements the following methods:</p>

- **Bank():** The constructor of the class initializes the array of account objects.
- **void createAccount(Account objAccount):** This method is invoked from the main method of the application. The new account holder's details are stored in the object of type Account. This object has been passed as a parameter to the createAccount() method. The method assigns a new account number by referring to the static variable lastAccountNumber. The method then writes the account details to a file using a binary stream. Use appropriate try catch blocks to handle all the possible exceptions that can be thrown due to the user inputs. If opening balance is < 100 dollars, then an error message is displayed to the user.
- **void withdraw():** This method is used to withdraw money from an account. This method accepts the account number and the amount to be withdrawn from the user. If the account number entered by the user is positive and the amount to be withdrawn is more than \$100 then data is read from the binary file. The data read from the binary file is stored in the array of accounts. The method then searches for the account number entered by the user in the array of accounts. A standard account in the Merchant's Bank must have a minimum account balance of \$500 at any given time. If the account has sufficient balance then the transaction proceeds and the updated account details are displayed to the user. Appropriate try catch blocks should be used to handle all the possible exceptions that can be thrown due to code.
- **void deposit():** This method is used to deposit money in an account. This method accepts the account number and the amount to be deposited from the user. If the account number entered by the user is positive and the amount to be deposited is more than \$100 then data is read from the binary file. The data read from the binary file is stored in the array of accounts. The method then searches for the account number entered by the user in the array of accounts. The account balance is updated if the account object corresponding to the account number is found. If the account number entered by the user doesn't match any of the accounts existing in the file then an error message is displayed to the user. Finally, the updated account details are displayed to the user. Appropriate try catch blocks should be used to handle all the possible exceptions that can be thrown due to code.
- **void display():** This method is used to display the details of a particular account. The method accepts the account number from the user. If the account number entered is positive then the method searches for the account record in the file. If the account exists then its details are displayed to the user. If the account does not exist then an error message is displayed to the user. Appropriate try catch blocks should be used to handle all the possible exceptions that can be thrown due to code.

BankTest.java

The BankTest class is a java main class used to test the Bank class. It creates an instance of the Bank class and displays the following menu of options to the user.

- Create a new account
- Withdraw Cash
- Deposit cash
- Display account details
- Exit

The user can select any of the options and a corresponding method is invoked on the instance of the Bank class. If the user chooses to create a new account then an object of the Account class is created and initialized using the inputs from the user. This object is then passed as an argument to the `createAccount()` method of the bank class. The application exits when the Exit option is selected.