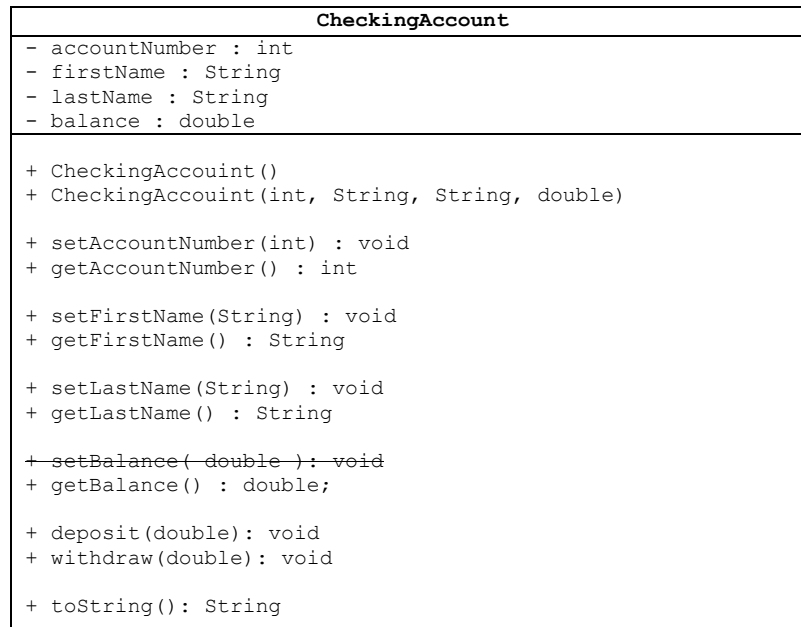


Do not use any data structures such as array or ArrayList to store data.

With a checking account, the customer can perform transactions such as deposit, withdrawal, and check balance. Use **object-oriented programming** approach, write a program to simulate the banking service.

- a) (CheckingAccount.java) Write the class definition CheckingAccount based on the following class diagram:



- Private instance variables: accountNumber, firstName, lastName, and balance.
 - Two constructors.
 - Public setters and getters for private instance data members.
 - Two public instance methods:
 - o deposit
 - It adds the deposit amount to the current balance.
 - Ensure a negative value is not added to the current balance.
 - o withdraw
 - It subtracts the withdraw amount from the current balance .
 - Ensure that the withdrawal amount does not exceed the current balance.
 - Ensure that a negative value is not withdrawn.
 - toString method.
 - Incorporate “**this**” reference in the constructors and setters.
- b) (HomeTownBank.java) Write a menu-driven client program that uses the CheckingAccount class to simulate the banking transactions.

Sample run:

```
*****Open Bank Account*****
```

```
Enter first name: Jon
Enter last name: Doe
Enter opening deposit: $500
```

Your new account has been created!

```
Account Number: 9001
Name: Jon Doe
Account Balance: $500.00
```

Welcome to "Home Town Bank"

```
*****
```

- 1 --- Deposit
- 2 --- Withdraw
- 3 --- Check balance
- 0 --- Exit

```
*****
```

Select an option: 1

```
Amount to deposit: $-200
Cannot deposit negative amount, try again.
```

```
*****
```

- 1 --- Deposit
- 2 --- Withdraw
- 3 --- Check balance
- 0 --- Exit

```
*****
```

Select an option: 1

Amount to deposit: \$200

New account balance: \$700.00

```
*****
```

- 1 --- Deposit
- 2 --- Withdraw
- 3 --- Check balance
- 0 --- Exit

```
*****
```

Select an option: 2

```
Amount to withdraw: $800
No enough cash to withdraw, try again.
```

```
*****
```

- 1 --- Deposit
- 2 --- Withdraw
- 3 --- Check balance
- 0 --- Exit

```
*****
```

Select an option: 2

Amount to withdraw: \$80

New account balance: \$620.00

```
*****
```

- 1 --- Deposit
- 2 --- Withdraw
- 3 --- Check balance
- 0 --- Exit

```
*****
```

```

Select an option: 3

Account Number: 9001
Name: Jon Doe
Account Balance: $620.00

*****
1 --- Deposit
2 --- Withdraw
3 --- Check balance
0 --- Exit
*****
Select an option: 6

Invalid selection (enter 1, 2, 3, or 0), please try again.

*****
1 --- Deposit
2 --- Withdraw
3 --- Check balance
0 --- Exit
*****
Select an option: 0

Thank you for banking with us!

```

Due: Wednesday, 11/23/22

- To receive full credit, the assignment must be submitted to D2L by the due date.
- Late submissions will incur a penalty of 5% per day.

Style, form, documentation, naming convention, and more

Each program should have a file header section. /* * Author: Your name * Date: Date of completion * Assignment: Assignment # NameOfSourceCode.java * Description: The program description */	Up to 5% deduction
Each program should be written with the appropriate form and style. Use indentation, blank line, and comments to make the source code easy to read.	Up to 5% deduction
Use Java naming convention and meaningful names to name the classes, methods, variables, constants, and other identifiers in the programs.	Up to 5% deduction
Format the output appropriately	Up to 5% deduction