THE PERSONAL ATM

Write a Java program which simulates a personal ATM. Your personal ATM should begin by presenting the following menu of options to the user upon start-up.

```
ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select >
```
Fig 1. The ATM Menu

About the Options

`B - Display Current Balance`
This option displays the current balance which starts out at $0.00.   When displaying the current balance, format the dollar amount to look like money, i.e., $100.00   (use the showAsDollars( ) method provided to accomplish this)

`D - Deposit`
This option requires that you build a static method called, deposit.   The deposit method will be called numerous times during the program.

> *static public double deposit(double amount, double currentBalance)*

`I - Add Monthly Interest`
This option requires that you build a static method called, addInterest( ).   The user can run this option once a month to deposit 1% to the current balance. Note: no interest is deposited if the balance is under $0.00.

> *static public double addInterest(double currentBalance)*

`T - Display Number of Transactions`
This option displays the number of transactions which have taken place. Include all of the deposits and withdrawals.

`W - Withdrawal`
This option requires that you build a static method called, withdrawal.   The withdrawal method allow an amount to be withdrawn from the account if and only if the currentBalance has enough funds to cover the withdrawal.

> *static public double withdrawal(double amount, double currentBalance)*

`L - Logout`
This option terminates the program.

Good Practice Guideline(s)

1. Static methods like deposit( ), withdrawal( ), and addInterest( ) should not output to the console. All outputting should be done by the main program. This practice allows the methods to be re-usable in other programs.

Sample Session

```
BlueJ: Terminal Window - ATM
 Options

ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > b
You've selected > B

Your balance is: $0.00


ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > d
You've selected > D

How much are you depositing > 100

ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > b
You've selected > B

Your balance is: $100.00
```

```
ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > w
You've selected > W

How much are you withdrawing > 0.50
Your withdrawal of $0.50 was successful.


ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > b
You've selected > B

Your balance is: $100.50


ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > t
You've selected > T

Number of Transactions:  3
```

```
ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > i
You've selected > I

Interest of $1.00 was added.


ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > b
You've selected > B

Your balance is: $101.00


ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > t
You've selected > T

Number of Transactions:  2

ATM Menu
B - Display Current Balance
D - Deposit
I - Add Monthly Interest
T - Display Number of Transactions
W - Withdrawal
L - Logout
Select > l
You've selected > L
```

Helper Method(s)

```
    /**
    * showAsDollars
    *
    * This routine formats the argument to look like dollars.
    *
    * This method accepts a double dollar amount and returns a string which is
    * formatted to look like dollars.
    *
    * @param  amount is the pre-formatted dollar amount  (e.g. 100.0)
    * @return a string which is formatted to look like dollars (e.g. $100.00)
    */
    public static String showAsDollars(double amount) {
          NumberFormat format = NumberFormat.getCurrencyInstance(Locale.CANADA);
          String currency = format.format(amount);
          return currency;
    }




    public static void main(String[] args)
    {//main

       //declarations


       double balance = 10.5;

                System.out.println( showAsDollars(balance) );   //will display as $10.50


    }//main
```
Fig 2. A method which formats a double to look like dollars, i.e. 10.5 would be displayed as $10.50