Ex.No: 7	**Bank Transaction System with User Exceptions**
Date:	

Aim:

To create a Java console application for Banking transaction system that helps the users to do their credit and debit transactions and it rises user defined exception while encountering errors in transaction and also it should be solved using exception handling techniques.

Algorithm:

```
Step 1  Start the Process
Step 2  Prompt user with List Operation Choices
        1. Add Money 2. Get Money 3. Details 4. Exit
        Get the choice from user.
Step 3  If user selects Add Money
        Step 3.1  Get the amount to be added to balance from the user
        Step 3.2  If the amount is less than 0 then throw Invalid Credit Exception and goto step
                  3.4
        Step 3.3  Else add amount to balance and goto step 2
        Step 3.4  Prompt user with “Enter valid credit amount” and goto step 3.1
Step 4  If user selects Get Money
        Step 4.1  Get the amount to be debited to balance from the user
        Step 4.2  If the amount is greater than existing balance then throw Invalid Debit
                  Exception and goto step 4.4
        Step 4.3  Else deduct amount from balance and goto step 2
        Step 4.4  Prompt user with “Enter valid debit amount” and goto step 4.1
Step 5  If user selects Details then display Customer Details [Name, Account Number, Current
        Balance]
Step 6  If user wants to exit display “Thank You !!!” and end process
Step 7  Stop the Process
```

Coding:

Customer.java

```
package com.raja.oopslab.exception.bank;

import java.util.Scanner;

public class Customer {
        String name;
        int accNo;
        int balance;

        public Customer(String name, int accNo) {
                this.name = name;
                this.accNo = accNo;
                this.balance = 0;
        }

        public void creditTransaction(int amount) {
                Scanner input = new Scanner(System.in);
                try {
                        if (amount < 0)
                                throw new InvalidCredit();

                        else
                                balance = balance + amount;
                } catch (InvalidCredit e) {
                        amount = input.nextInt();
                        creditTransaction(amount);
                }

        }

        public void debitTransaction(int amount) {
                Scanner input = new Scanner(System.in);
                try {
                        if (amount > balance)
                                throw new InvalidDebit();

                        else
                                balance = balance - amount;
                } catch (InvalidDebit e) {
                        amount = input.nextInt();
                        debitTransaction(amount);
                }

        }

        public void displayDetails(){
                System.out.println("Customer Details");
                System.out.println("****************");
```

```
		System.out.println("Customer Name : "+this.name);
		System.out.println("Customer AccNo : "+this.accNo);
		System.out.println("Customer Current Balance : "+this.balance);
	}

}
```

InvalidCredit.java

```
package com.raja.oopslab.exception.bank;

public class InvalidCredit extends Exception {
	public InvalidCredit() {
		System.out.print("Please enter valid credit amount");
	}
}
```

InvalidDebit.java

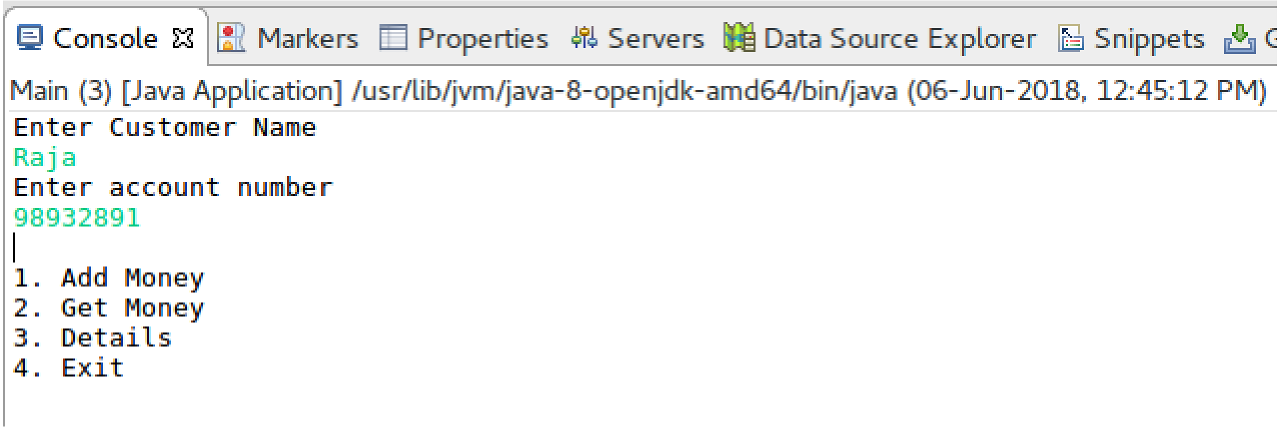
```
package com.raja.oopslab.exception.bank;

public class InvalidDebit extends Exception {
	public InvalidDebit() {
		System.out.print("Please enter valid debit amount");
	}
}
```

Main.java

```
import java.util.Scanner;
import com.raja.oopslab.exception.bank.*;
public class Main {
        public static void main(String[] args) {
                Scanner input = new Scanner(System.in);
                String name;
                int acc_no;
                System.out.println("Enter Customer Name");
                name = input.next();
                System.out.println("Enter account number");
                acc_no = input.nextInt();
                Customer aCustomer = new Customer(name, acc_no);
                int choice = 0;
                while(choice != 4){
                        System.out.println("\n1. Add Money\n2. Get Money\n3. Details\n4. Exit");
                        choice = input.nextInt();
                        switch(choice){
                        case 1:
                                System.out.println("Enter the amount");
                                aCustomer.creditTransaction(input.nextInt());
                                break;
                        case 2:
                                System.out.println("Enter the amount");
                                aCustomer.debitTransaction(input.nextInt());
                                break;
                        case 3:
                                aCustomer.displayDetails();
                                break;
                        case 4:
                                System.out.println("Thank You !!!");
                                break;
                        }
                }
        }
}
```

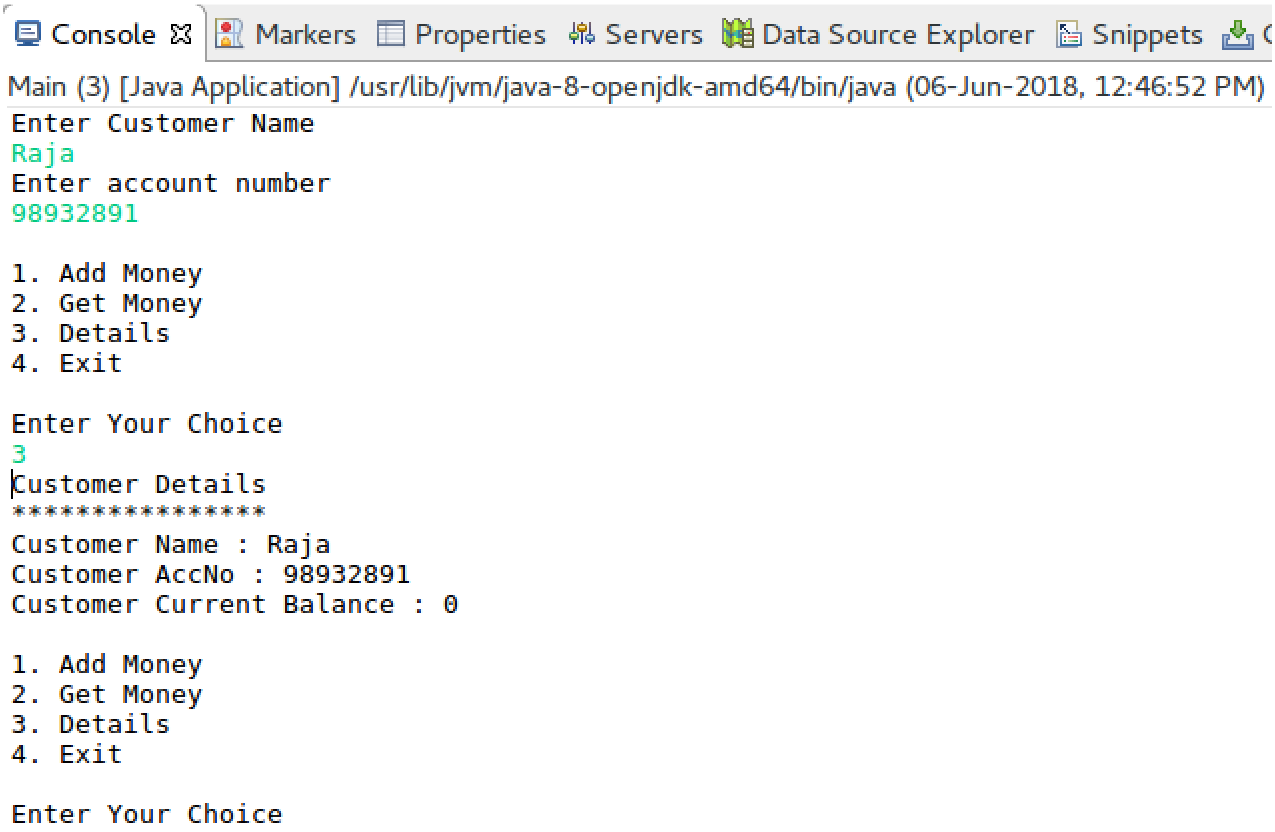
Output:

Choice:



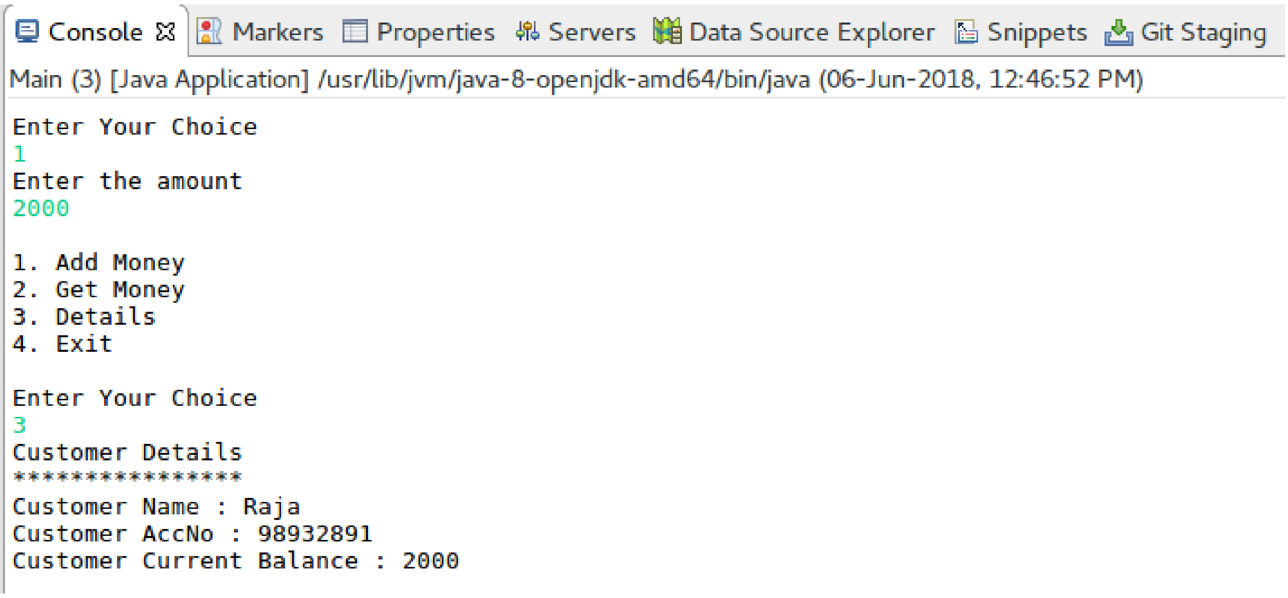
```
Main (3) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (06-Jun-2018, 12:45:12 PM)
Enter Customer Name
Raja
Enter account number
98932891

1. Add Money
2. Get Money
3. Details
4. Exit
```

Display Details:



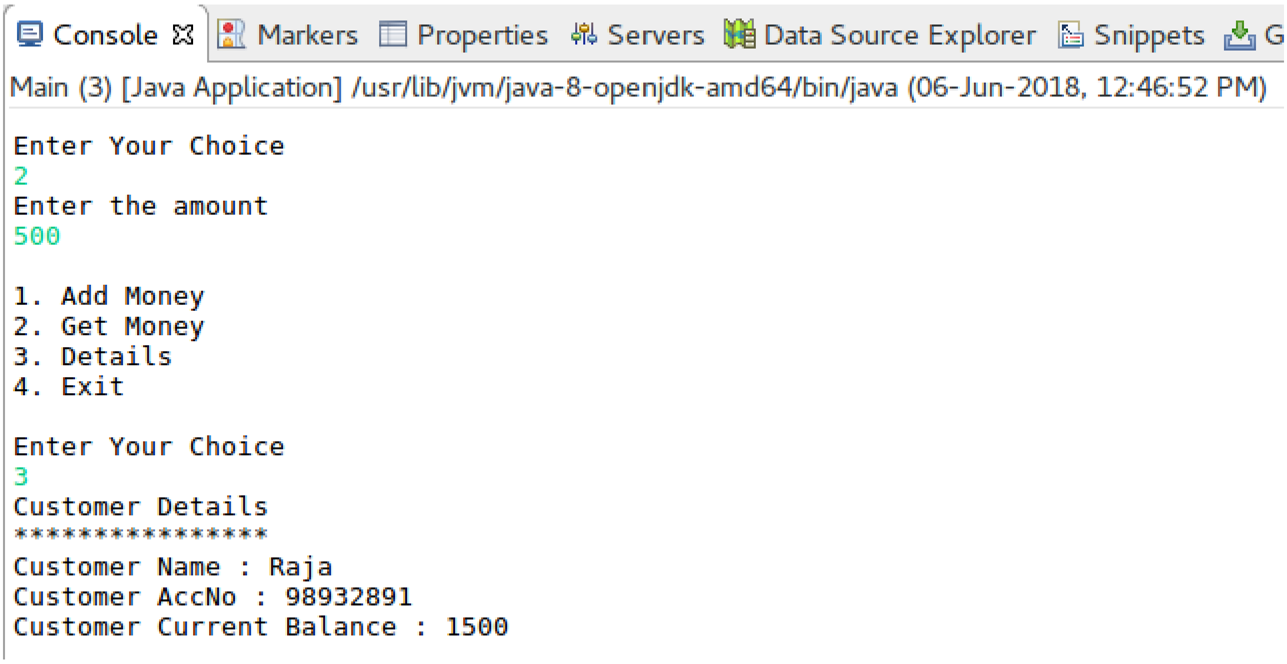
```
Main (3) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (06-Jun-2018, 12:46:52 PM)
Enter Customer Name
Raja
Enter account number
98932891

1. Add Money
2. Get Money
3. Details
4. Exit

Enter Your Choice
3
Customer Details
****************
Customer Name : Raja
Customer AccNo : 98932891
Customer Current Balance : 0

1. Add Money
2. Get Money
3. Details
4. Exit

Enter Your Choice
```

Add Money:



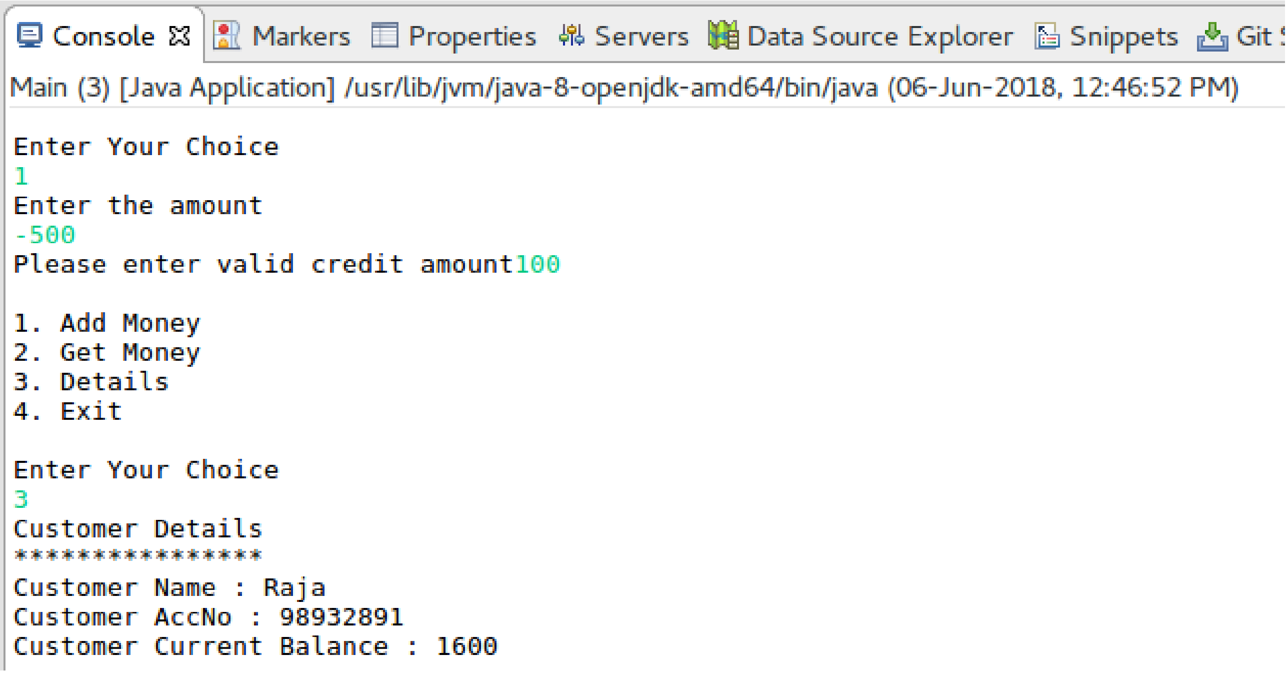
```
Main (3) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (06-Jun-2018, 12:46:52 PM)
Enter Your Choice
1
Enter the amount
2000

1. Add Money
2. Get Money
3. Details
4. Exit

Enter Your Choice
3
Customer Details
****************
Customer Name : Raja
Customer AccNo : 98932891
Customer Current Balance : 2000
```

Get Money:



```
Main (3) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (06-Jun-2018, 12:46:52 PM)
Enter Your Choice
2
Enter the amount
500

1. Add Money
2. Get Money
3. Details
4. Exit

Enter Your Choice
3
Customer Details
****************
Customer Name : Raja
Customer AccNo : 98932891
Customer Current Balance : 1500
```

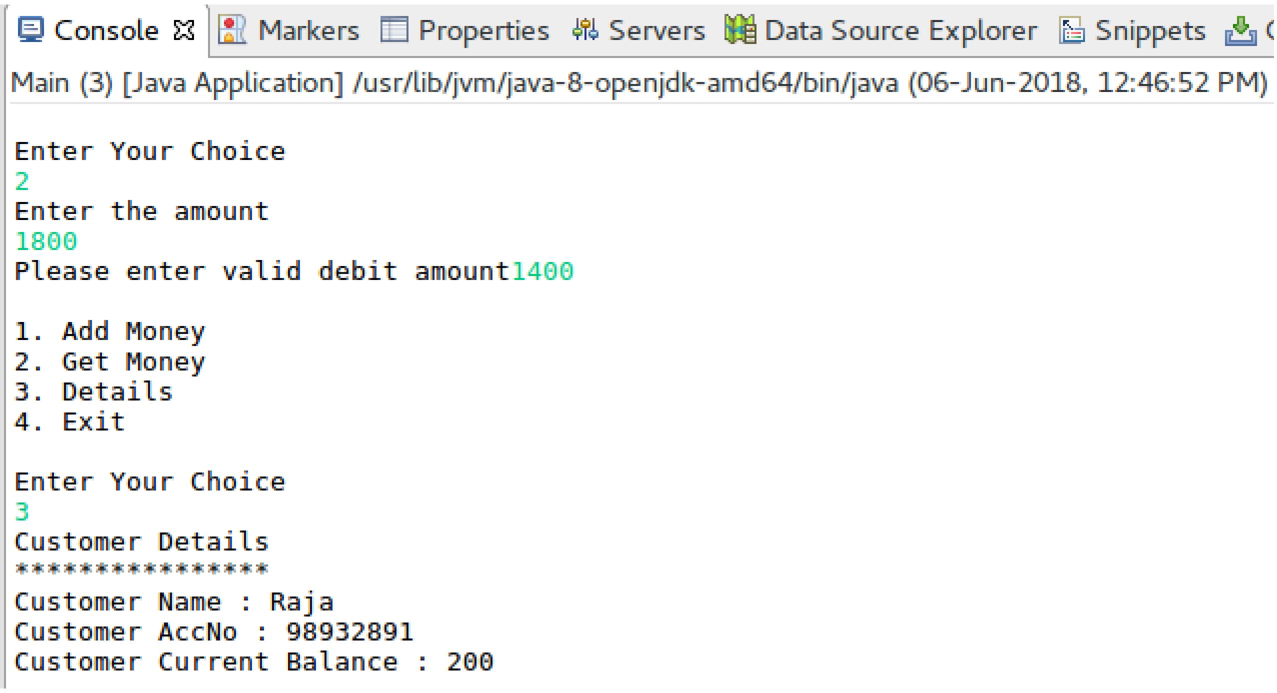
Exception in Add Money:



Main (3) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (06-Jun-2018, 12:46:52 PM)

```
Enter Your Choice
1
Enter the amount
-500
Please enter valid credit amount100

1. Add Money
2. Get Money
3. Details
4. Exit

Enter Your Choice
3
Customer Details
****************
Customer Name : Raja
Customer AccNo : 98932891
Customer Current Balance : 1600
```

Exception in Get Money:



Main (3) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (06-Jun-2018, 12:46:52 PM)

```
Enter Your Choice
2
Enter the amount
1800
Please enter valid debit amount1400

1. Add Money
2. Get Money
3. Details
4. Exit

Enter Your Choice
3
Customer Details
****************
Customer Name : Raja
Customer AccNo : 98932891
Customer Current Balance : 200
```

Result:

The java console application that uses user defined exception handling techniques was developed and tested successfully.