Project

“Computer Science and Engineering”

Lovely Professional University

“Bank Management System”

Submitted to-

Amarinder Kaur mam

Submitted by –

Name	**Reg. No**	**Roll No.**
Ashok	12113114	Rk21PKA22
Prakash Bhati	12109963	Rk21PKB60
Bhumika Shinde	12200124	Rk21PKA23

“Bank Management System”

Introduction : The main objective of the project is to develop online Banking system for banks. In present system all banking work is done manually. User have to visit bank to Withdrawal or Deposit amount. In present bank system it is also difficult to find account information of account holder. In this bank management system we will automate all the banking process. It creates a class called "BankAccount" which has attributes like balance, previous transaction, customer name, and customer ID. The class has methods for deposit, withdrawal, getting previous transactions, and displaying a menu for the user to interact with the system.

The "BankAccount" class is initialized with a customer name and ID when it is created. The "deposit" method allows the user to add funds to their account, while the "withdraw" method allows them to withdraw funds if they have sufficient balance. The "getPreviousTransaction" method displays the user's previous transaction, and the "showMenu" method displays a menu of options for the user to interact with the system.

The "BankingSystem" class contains the main method which creates an instance of the "BankAccount" class with a customer name and ID and then calls the "showMenu" method to start the interaction between the user and the banking system.

Modules of project:

- **Scanner:** A class that reads user input from the console.
- **System:** A class that provides access to the system resources, such as standard input and output.
- **InputMismatchException:** A class that represents an exception thrown when the input provided by the user is of the wrong type.
- **Exception handling:** A class that represents a generic exception.
  It is important to handle exceptions when working with databases, as unexpected errors can occur during the execution

Features (concepts) of java used in project:

Some of the features of Java used in the given code for a bank management system project are:

I. **Classes and Objects:** The code defines a class called BankAccount, which is used to create objects for each bank account.

II. **Constructors:** The code defines a constructor in the BankAccount class, which is used to initialize the customerName and customerId fields.

III. **Variables:** The code uses variables like balance, previousTransaction, customerName, and customerId to store data.

IV. **Methods:** The code defines several methods in the BankAccount class, such as deposit(), withdraw(), getPreviousTransaction(), and showMenu().

V. **Conditional Statements**: The code uses if-else statements to check whether a withdrawal amount is greater than the available balance.

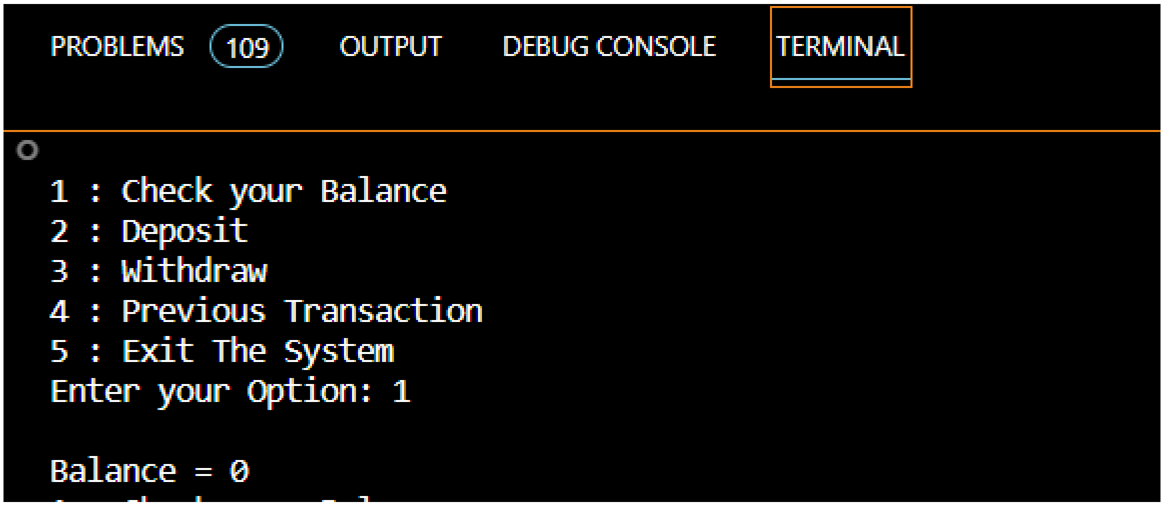
VI. **Switch Statements:** The code uses switch statements to handle the user's menu options.

VII. **Exception Handling:** The code uses try-catch blocks to handle exceptions like InputMismatchException and other general exceptions.

VIII. **Input and Output:** The code uses the Scanner class to get input from the user and System.out.println() to display output to the user.

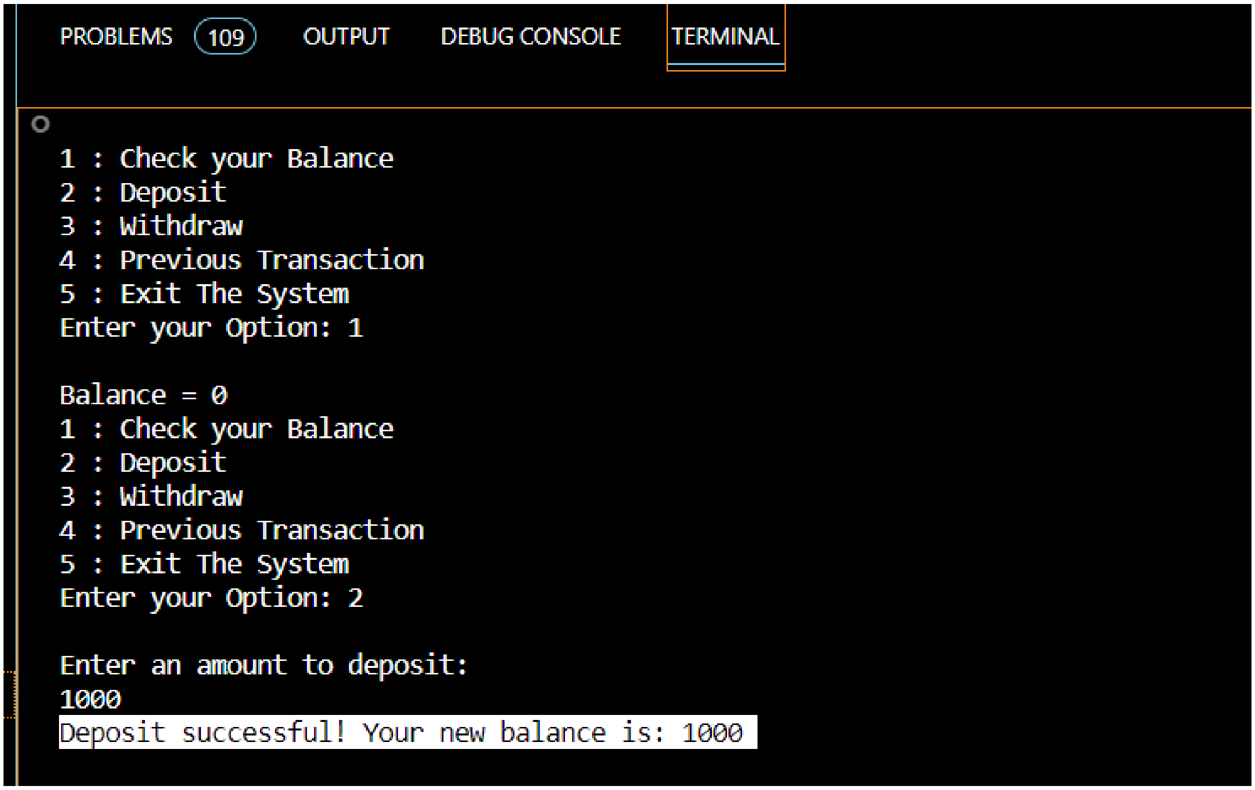
Screen shots of output:

1.Check Balance:



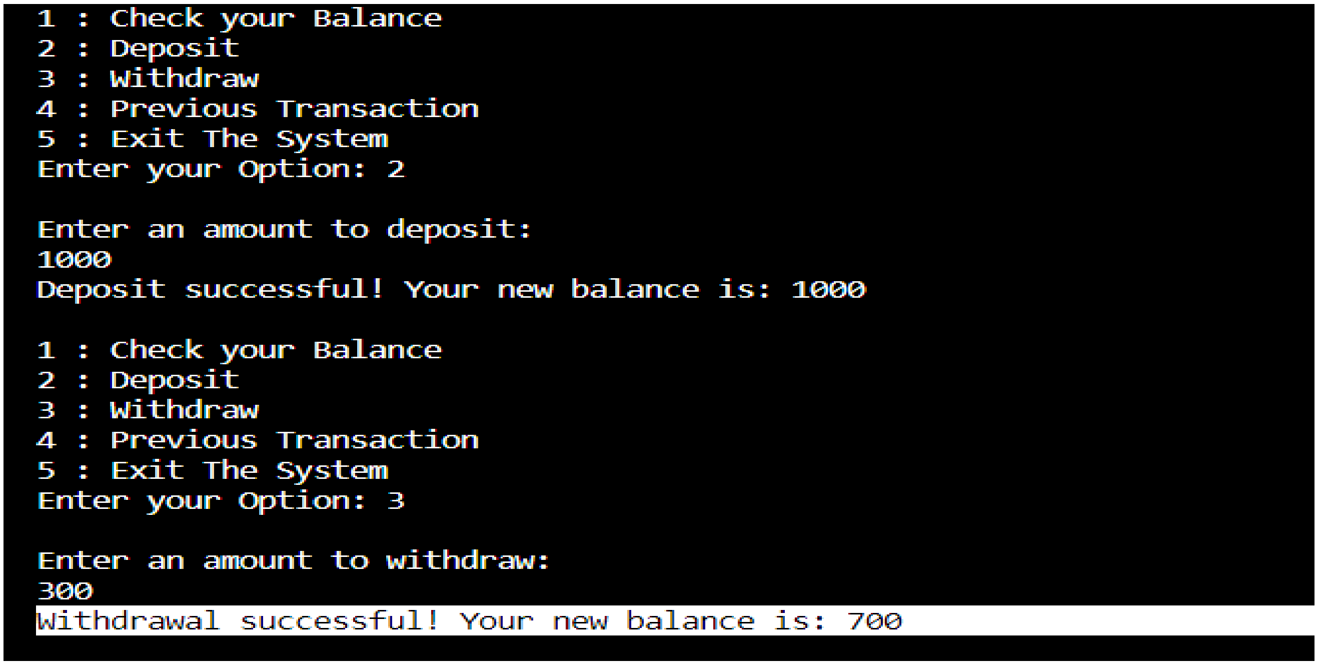
```
1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 1

Balance = 0
```

2 : Deposit:



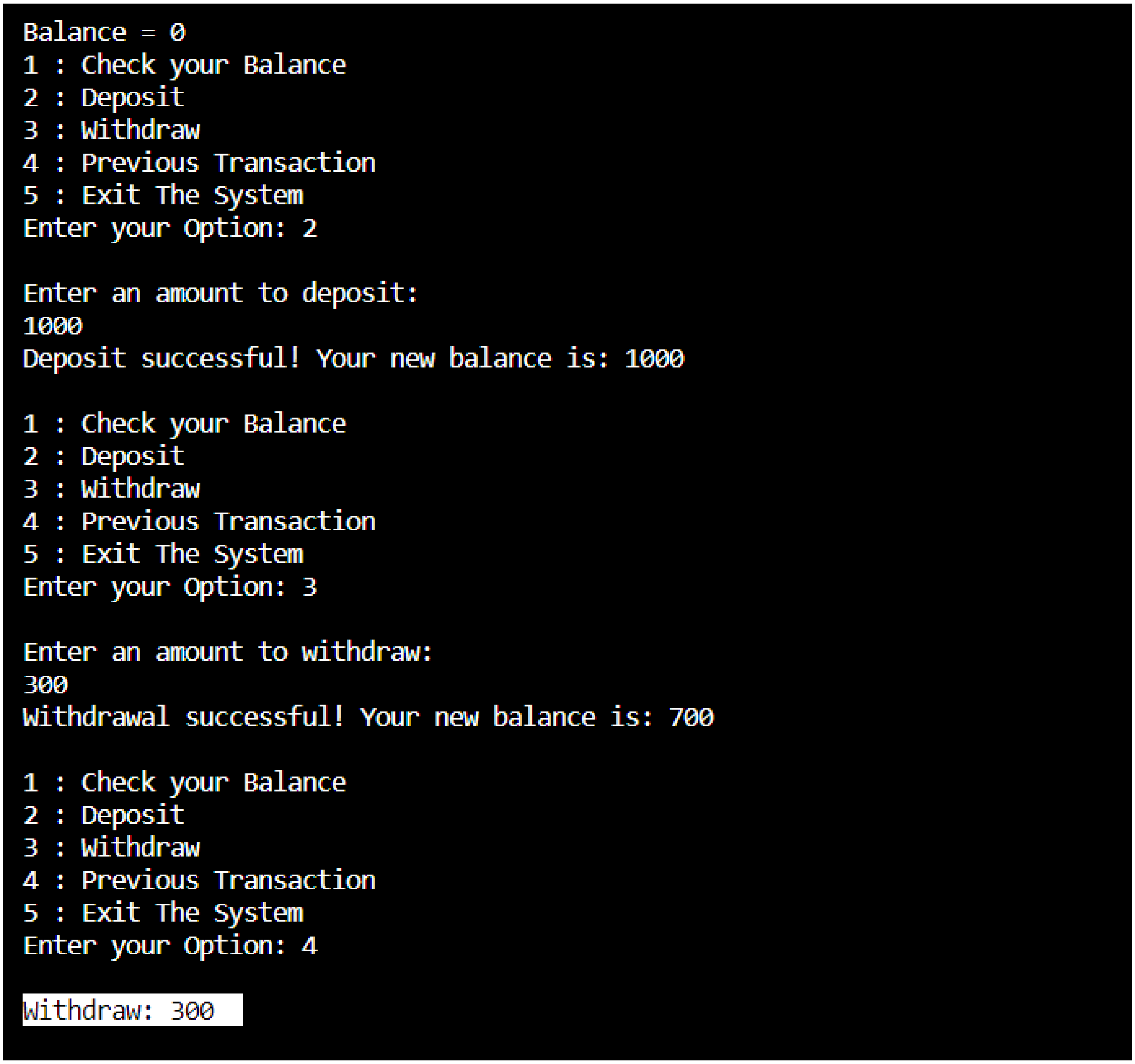
```
1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 1

Balance = 0
1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 2

Enter an amount to deposit:
1000
Deposit successful! Your new balance is: 1000
```

3 : Withdraw:



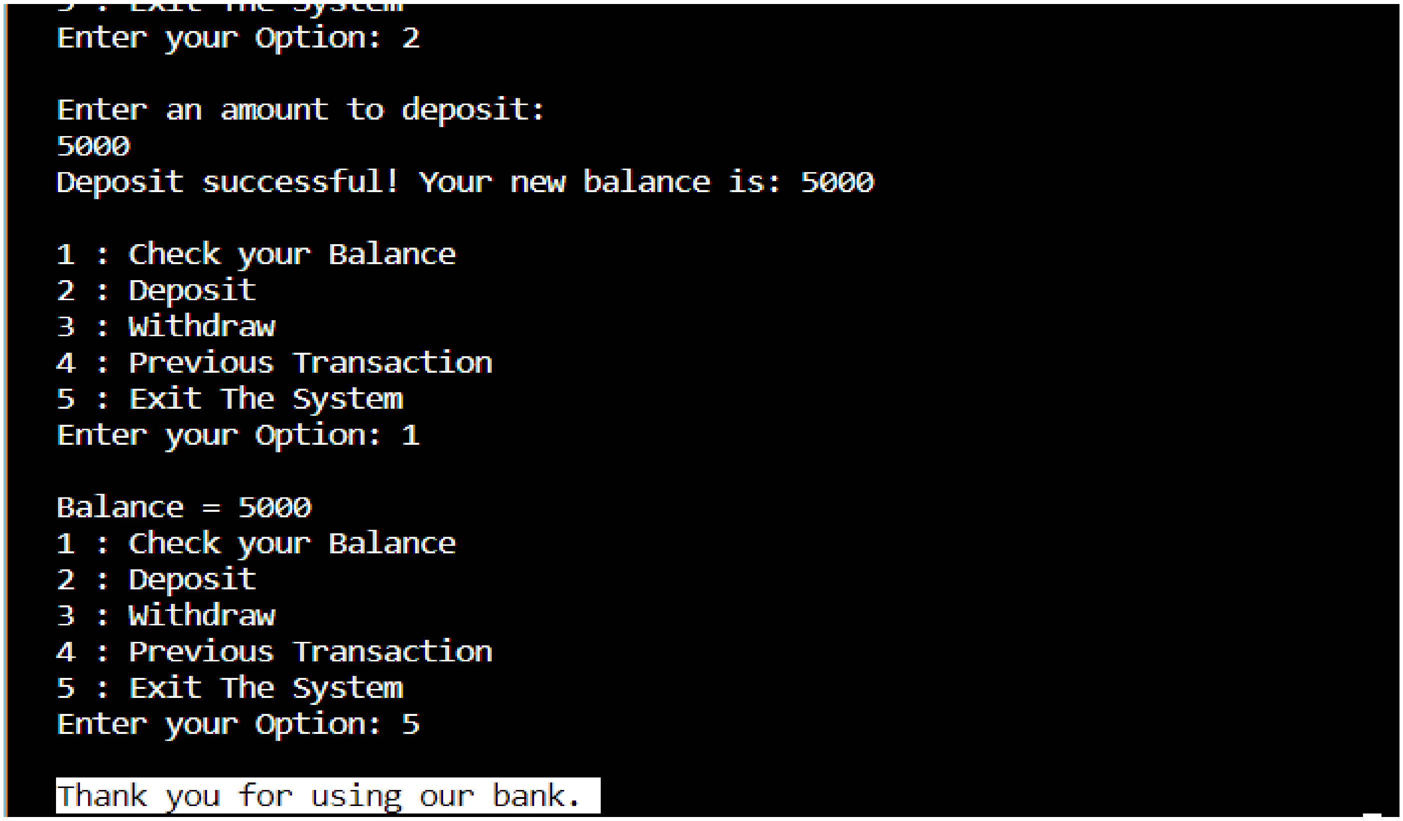
```
1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 2

Enter an amount to deposit:
1000
Deposit successful! Your new balance is: 1000

1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 3

Enter an amount to withdraw:
300
Withdrawal successful! Your new balance is: 700
```

4 : Previous Transaction:



```
Balance = 0
1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 2

Enter an amount to deposit:
1000
Deposit successful! Your new balance is: 1000

1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 3

Enter an amount to withdraw:
300
Withdrawal successful! Your new balance is: 700

1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 4

Withdraw: 300
```

5 : Exit The System:



```
5 : Exit The System
Enter your Option: 2

Enter an amount to deposit:
5000
Deposit successful! Your new balance is: 5000

1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 1

Balance = 5000
1 : Check your Balance
2 : Deposit
3 : Withdraw
4 : Previous Transaction
5 : Exit The System
Enter your Option: 5

Thank you for using our bank.
```

Work Done by each Team Member:

Ashok(12113114):-

1. Exception Handling part
    a. InputMismatchException
    b. Exception

2. getPreviousTransaction() method

Prakash Bhati(12109963):-

1. Withdraw() method part
2. Constructor part

Bhumika Shinde(12200124):-

1. Deposit() method part
2. Conditional-statements part
    a. Switch Case part

GitHub link:-

https://github.com/Prakashbhati086/Bank_management_System_60_22_23.git

Thank You