

**Sarhad University of Science and Information Technology,**

**Name:- Arman Khan**

**Reg:-SU-21-01-001-001**

**Department:-BSCS**

**Subject:-JAVA**

**Semester:-5th**

**Submitted To:-Engr Ayub Ashraf**

# Bank Application Project Documentation

## Overview

The Bank Application Project is a simple command-line Java application that simulates basic banking operations. It allows users to manage their bank accounts by checking balances, depositing money, withdrawing money, viewing previous transactions, and exiting the system.

## Classes

### 1. BankAccount

* **Description**: Represents a bank account with attributes such as balance, previous transaction amount, customer name, and customer ID. Provides methods to perform banking operations.
* **Attributes**:
  + **double bal**: Stores the current balance of the account.
  + **double prevTrans**: Stores the amount of the previous transaction.
  + **String customerName**: Stores the name of the customer.
  + **String customerId**: Stores the ID of the customer.
* **Methods**:
  + **BankAccount(String customerName, String customerId)**: Constructor to initialize customer name and ID.
  + **void deposit(double amount)**: Deposits the specified amount into the account.
  + **void withdraw(double amt)**: Withdraws the specified amount from the account.
  + **void getPreviousTrans()**: Displays information about the previous transaction.
  + **void menu()**: Displays a menu of options and handles user interaction.

### 2. BankApplicationProject

* **Description**: Contains the main method to run the application.
* **Methods**:
  + **main(String[] args)**: Entry point of the application. Prompts the user to enter their name and customer ID, creates a **BankAccount** object, and calls the **menu()** method to start the banking operations.

## Usage

1. Run the **BankApplicationProject** class.
2. Enter your name and customer ID when prompted.
3. Choose an option from the menu to perform banking operations:
   * **a**: Check Balance
   * **b**: Deposit Amount
   * **c**: Withdraw Amount
   * **d**: Previous Transaction
   * **e**: Exit
4. Follow the instructions for each option to perform the desired operation.
5. Repeat steps 3-4 until you choose to exit the system.

## Limitations and Future Improvements

* The application is command-line based and lacks a graphical user interface (GUI), which may not provide a user-friendly experience.
* Error handling is minimal. Additional checks and validation could be added to handle edge cases and invalid user inputs more gracefully.
* The application does not persist data between sessions. Implementing data storage and retrieval mechanisms, such as a database or file system, could enhance functionality.