BANKING SYSTEM-OOPS,COLLECTIONS AND EXCEPTION HANDLING

TASK 9: ABSTRACTION

- BankAccount is an abstract class that holds common attributes (accountNumber, customerName, balance) and abstract methods for deposit, withdrawal, and interest calculation.

- SavingsAccount extends BankAccount and adds an interestRate. It allows deposits, withdrawals (only if funds are sufficient), and calculates interest based on the balance.

- CurrentAccount extends BankAccount with an overdraft feature (e.g., ₹5000 limit). It allows withdrawals even beyond the balance up to the limit, but no interest is applied.

- Bank class manages the account creation and operations via a menu-driven interface using switch-case, letting the user choose between savings or current account.

- Operations include deposit, withdraw, interest calculation (only for savings), and displaying account info—all handled interactively through console input/output.

BANKACCOUNT.JAVA

```
abstract class BankAccount {
   protected String accountNumber;
   protected String customerName;
   protected float balance;

   public BankAccount() {
      this.accountNumber = "";
      this.customerName = "";
      this.balance = 0.0f;
   }

   public BankAccount(String accountNumber, String customerName, float balance) {
      this.accountNumber = accountNumber;
      this.customerName = customerName;
      this.balance = balance;
   }

   public String getAccountNumber() { return accountNumber; }
   public void setAccountNumber(String accountNumber) { this.accountNumber =
accountNumber; }

   public String getCustomerName() { return customerName; }
   public void setCustomerName(String customerName) { this.customerName =
customerName; }

   public float getBalance() { return balance; }
   public void setBalance(float balance) { this.balance = balance; }
```

```
    public void printAccountDetails() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Balance: " + balance);
    }

    public abstract void deposit(float amount);
    public abstract void withdraw(float amount);
    public abstract void calculateInterest();
}
```

CURRENTACCOUNT.JAVA

```
class CurrentAccount extends BankAccount {
    private static final float OVERDRAFT_LIMIT = 5000.0f;

    public CurrentAccount(String accountNumber, String customerName, float balance) {
        super(accountNumber, customerName, balance);
    }

    @Override
    public void deposit(float amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

    @Override
    public void withdraw(float amount) {
        if (amount <= (balance + OVERDRAFT_LIMIT)) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Overdraft limit exceeded.");
        }
    }

    @Override
    public void calculateInterest() {
        System.out.println("No interest for current accounts.");
    }
}
```

BANK.JAVA

```
import java.util.Scanner;

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        BankAccount account = null;

        System.out.println("==== Welcome to HexaBank ====");
        System.out.println("1. Create Savings Account");
        System.out.println("2. Create Current Account");
        System.out.print("Enter your choice: ");
        int choice = sc.nextInt();
        sc.nextLine();  // Consume newline

        // Common input
        System.out.print("Enter Account Number: ");
        String accNo = sc.nextLine();
        System.out.print("Enter Customer Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Initial Balance: ");
        float balance = sc.nextFloat();

        switch (choice) {
            case 1:
                System.out.print("Enter Interest Rate (%): ");
                float rate = sc.nextFloat();
                account = new SavingsAccount(accNo, name, balance, rate);
                break;
            case 2:
                account = new CurrentAccount(accNo, name, balance);
                break;
            default:
                System.out.println("Invalid option!");
                System.exit(0);
        }

        int option;
        do {
            System.out.println("\n--- Operations Menu ---");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Calculate Interest");
            System.out.println("4. Show Account Details");
            System.out.println("5. Exit");
            System.out.print("Choose option: ");
            option = sc.nextInt();

            switch (option) {
```

```
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    float dep = sc.nextFloat();
                    account.deposit(dep);
                    break;
                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    float with = sc.nextFloat();
                    account.withdraw(with);
                    break;
                case 3:
                    account.calculateInterest();
                    break;
                case 4:
                    account.printAccountDetails();
                    break;
                case 5:
                    System.out.println("Thank you for banking with us!");
                    break;
                default:
                    System.out.println("Invalid choice!");
            }

        } while (option != 5);

        sc.close();
    }
}
```

OUTPUT:

```
1. Create Savings Account
2. Create Current Account
Enter your choice: 1
Enter Account Number: 5001
Enter Customer Name: Praveshini
Enter Initial Balance: 1000
Enter Interest Rate (%): 5

--- Operations Menu ---
1. Deposit
2. Withdraw
3. Calculate Interest
4. Show Account Details
5. Exit
Choose option: 1
Enter amount to deposit: 5000
Deposited: 5000.0

--- Operations Menu ---
1. Deposit
2. Withdraw
3. Calculate Interest
4. Show Account Details
5. Exit
```

```
Choose option: 2
Enter amount to withdraw: 1000
Withdrawn: 1000.0

--- Operations Menu ---
1. Deposit
2. Withdraw
3. Calculate Interest
4. Show Account Details
5. Exit
Choose option: 3
Interest added: 250.0

--- Operations Menu ---
1. Deposit
2. Withdraw
3. Calculate Interest
4. Show Account Details
5. Exit
Choose option: 4
Account Number: 5001
Customer Name: Praveshini
Balance: 5250.0
```

```
--- Operations Menu ---
1. Deposit
2. Withdraw
3. Calculate Interest
4. Show Account Details
5. Exit
Choose option: 5
Thank you for banking with us!
```