BANKING SYSTEM-OOPS,COLLECTIONS AND EXCEPTION HANDLING

TASK 7:CLASS & OBJECT

ACCOUNT CLASS

- Account details – Stores account number, type (Savings/Current), and balance.
- Getter & Setter Methods – Provides controlled access to private variables.
- Deposit & Withdraw Methods – Allows adding and withdrawing money with validation.
- Interest Calculation – Adds 4.5% interest for Savings accounts only.
- Print Method – Displays account details clearly.

```
public class Account {
    private int accountNumber;
    private String accountType;
    private double balance;

    public Account() {}

    public Account(int accountNumber, String accountType, double balance) {
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

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

    public String getAccountType() {
        return accountType;
    }
    public void setAccountType(String accountType) {
        this.accountType = accountType;
    }

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

```
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount!");
        }
    }
    public void withdraw(double amount) {
        if (amount > 0 && balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance or invalid amount!");
        }
    }
    public void calculateInterest() {
        if (accountType.equalsIgnoreCase("Savings")) {
            double interest = balance * 0.045;
            balance += interest;
            System.out.println("Interest added: " + interest);
        } else {
            System.out.println("Interest calculation only applies to Savings accounts.");
        }
    }

    public void printAccountInfo() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: " + balance);
    }
}
```

CUSTOMER CLASS

- Customer details – Stores customer ID, name, email, phone, and address.
- Getter & Setter Methods – Controls access to private customer attributes.
- Constructor Overloading – Supports object creation with or without initial values.
- Print Method – Displays customer details.

```
public class Customer {
    private int customerId;
    private String firstName;
    private String lastName;
```

```
    private String email;
    private String phoneNumber;
    private String address;

    public Customer() {}

    public Customer(int customerId, String firstName, String lastName, String email, String
phoneNumber, String address) {
        this.customerId = customerId;
        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
        this.phoneNumber = phoneNumber;
        this.address = address;
    }

    public int getCustomerId() {
        return customerId;
    }
    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }

    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }
    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }
```

```
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }

    public void printCustomerInfo() {
        System.out.println("Customer ID: " + customerId);
        System.out.println("Name: " + firstName + " " + lastName);
        System.out.println("Email: " + email);
        System.out.println("Phone: " + phoneNumber);
        System.out.println("Address: " + address);
    }
}
```

BANK CLASS

- Handles Input & Output – Uses Scanner to get user input.
- Creates Customer & Account Objects – Instantiates Customer and Account classes.
- Performs Transactions – Calls deposit and withdraw functions.
- Applies Interest – Triggers interest calculation for Savings accounts.
- Displays Final Account State – Prints updated balance after all operations.

```
import java.util.*;
public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Customer ID: ");
        int customerId = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter First Name: ");
        String firstName = scanner.nextLine();
        System.out.print("Enter Last Name: ");
        String lastName = scanner.nextLine();
        System.out.print("Enter Email: ");
        String email = scanner.nextLine();
        System.out.print("Enter Phone: ");
        String phone = scanner.nextLine();
        System.out.print("Enter Address: ");
        String address = scanner.nextLine();

        Customer customer = new Customer(customerId, firstName, lastName, email, phone,
address);
```

```
        customer.printCustomerInfo();

        System.out.print("Enter Account Number: ");
        int accountNumber = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Account Type (Savings/Current): ");
        String accountType = scanner.nextLine();
        System.out.print("Enter Initial Balance: ");
        double balance = scanner.nextDouble();

        Account account = new Account(accountNumber, accountType, balance);
        account.printAccountInfo();

        System.out.print("Enter Deposit Amount: ");
        double depositAmount = scanner.nextDouble();
        account.deposit(depositAmount);
        System.out.print("Enter Withdraw Amount: ");
        double withdrawAmount = scanner.nextDouble();
        account.withdraw(withdrawAmount);
        account.calculateInterest();
        account.printAccountInfo();
    }
}
```

OUTPUT:

```
Enter Customer ID: 1
Enter First Name: Praveshini
Enter Last Name: BNV
Enter Email: praveshini@gmail.com
Enter Phone: 9994650511
Enter Address: Madurai
Customer ID: 1
Name: Praveshini BNV
Email: praveshini@gmail.com
Phone: 9994650511
Address: Madurai
Enter Account Number: 5000
Enter Account Type (Savings/Current): Savings
Enter Initial Balance: 5000
Account Number: 5000
Account Type: Savings
Balance: 5000.0
Enter Deposit Amount: 2000
Deposited: 2000.0
Enter Withdraw Amount: 1000
Withdrawn: 1000.0
Interest added: 270.0
Account Number: 5000
Account Type: Savings
Balance: 6270.0
BUILD SUCCESSFUL (total time: 2 minutes 12 seconds)
```