PPL Unit 3 Case Study-classes, objects data and methods -online banking (In Java)

Why OOP?:

1. The application has a number of components, using OOP allows us to break down our software into smaller components.
2. OOP makes troubleshooting easier for these types of softwares.
3. Concepts like access specifiers will enable data protection.

Class *UserAccount* contains the following attributes for an object:
Account_number, username, password, balance;
where in password and balance need to be private
Along with getter and setters the following methods will be required:
For initialising a new account, verifying transaction, performing the transaction, returning balance.

Class *Bank* contains the following attributes for an object:
List of accounts,customer count;
The methods required would be :
For creating a new account, initialising the transaction, displaying account details

Following is the code with an output:

```
import java.util.*;
import java.io.*;
class UserAccount{
    String name;
    private String password;
    private double balance;
    UserAccount(){
        Scanner scan = new Scanner(System.in);
        System.out.print("Name: ");
        while(!scan.hasNext()){}
        name = scan.nextLine();
        System.out.print("Password: ");
        while(!scan.hasNext()){}
        password=scan.nextLine();
        balance=0;
        System.out.println("Created Successfully");
    }
    Boolean verifyuser(){
        Scanner scan = new Scanner(System.in);
        System.out.print("Password to verify: ");
        while(!scan.hasNext()){}
        String pass=scan.nextLine();
        return password.equals(pass);
    }
    Boolean transact(double amt){
        if(verifyuser()){
            balance+=amt;
            System.out.println("Transaction successful!");
            return true;
        }else{
            System.out.println("Wrong password!");
            System.out.println("Transaction unsuccessful!");
            return false;
        }
    }
    double getbalance(){
        return balance;
    }
}
class Bank{
    int customercount=0;
    UserAccount[] accounts=new UserAccount[10];
    void create(){
        accounts[customercount]=new UserAccount();
        customercount++;
    }
    void performT(){
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter Amount: ");
        while(!scan.hasNext()){}
        double b=scan.nextDouble();
        System.out.print("acc id: ");
        while(!scan.hasNext()){}
        int id=scan.nextInt();
        accounts[id].transact(b);
    }
    void display(int id){
        System.out.println(id);
        System.out.println(accounts[id].name);
        System.out.println(accounts[id].getbalance());
        System.out.println();
    }
    void displayAll(){
        for(int i=0;i<customercount;i++){
```

```
      display(i);
    }
  }
}
class Main {
 public static void main(String[] args) {
  int c;Bank bank=new Bank();
  Scanner scan = new Scanner(System.in);
  while(true){
    System.out.println("What do you want to do?");
    System.out.println("1. Create new UserAccount");
    System.out.println("2. Perform a transaction");
    System.out.println("3. Display my account");
    System.out.println("4. Display all accounts");
    System.out.println("0. Exit");
    System.out.println();
    while(!scan.hasNext()){}
    c=scan.nextInt();
    if(c==0){ scan.close();break;}
    switch(c){
      case 1: bank.create();break;
      case 2: bank.performT();break;
      case 3: System.out.print("account id:
");while(!scan.hasNext()){}c=scan.nextInt();bank.display(c);break;
      case 4: bank.displayAll();break;
      default:;
    }
  }
  System.out.println("Done!");
 }
/*
}
Sample tested:

What to perform?
1. Create new UserAccount
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit

1
Name: Suyog
Password: suyog@onlinebanking
```

```
Created Successfully
What to perform?
1. Create new UserAccount
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit

1
Name: Alex
Password: alex@onlinebanking
Created Successfully
What to perform?
1. Create new UserAccount
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit

1
Name: Bob
Password: bob@onlinebanking
Created Successfully
What to perform?
1. Create new UserAccount
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit

1
Name: Vasya
Password: vasya@onlinebanking
Created Successfully
What to perform?
1. Create new UserAccount
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit

2
Enter Amount: 1234
acc id: 0
```

```
Password to verify: suyog@onlinebanking
Transaction successful!
What to perform?
1. Create new UserAccount
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit

2
Enter Amount: 123
acc id: 1
Password to verify: alex@onlinebanking
Transaction successful!
What to perform?
1. Create new UserAccount
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit

2
Enter Amount: 12345
acc id: 2
Password to verify: vasya@onlinebanking
Wrong password!
Transaction unsuccessful!
What to perform?
1. Create new UserAccount
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit


3
account id: 1
1
Suyog
123.0

What to perform?
1. Create new UserAccount
2. Perform transaction
```

```
3. Display my account
4. Display all accounts
0. Exit

4
0
Suyog
1234.0

1
Alex
123.0

2
Bob
12345.0

3
Vasya
0.0
What to perform?
1. Create new Account
2. Perform transaction
3. Display my account
4. Display all accounts
0. Exit
0
Done!
*/
```