

Q.

:- Lab - 12 :-

→ Account:-

```
public class Account {  
    public String firstName;  
    public String lastName;  
    double balance;  
    public Branch branch;  
    public int AccountType;  
  
    public String toString() {  
        return "Current balance is " + balance;  
    }  
  
    public Branch getBranch() {  
        return branch;  
    }  
}
```

## \* Checking Account:—

```
public class CheckingAccount extends Account {
    public CheckingAccount (String firstname, String lastname,
        Branch b) {
```

```
        this.firstname = firstname;
```

```
        this.lastname = lastname;
```

```
        this.branch = b;
```

```
        this.accounttype = 1;
```

```
    }
```

```
    public String getDescription () {
```

```
        return "Customer: "
```

```
            + lastname + ", "
```

```
            + firstname + "V.B. Branch: "
```

```
            + branch + ", " + branch.getserviceloc();
```

```
    }
```

```
    public double getbalance () { return balance; }
```

```
    public void deposit (double amount) { balance += amount; }
```

```
    public void withdraw (double amount) { balance -= amount; }
```

```
    public String toString () { return this.getDescription() +
```

```
        "Checking balance is " + balance; }
```

```
}
```



\* Saving Account :- same as checking account.  
but Saving Account type = 2;

\* Bank Interface :-

```
public interface BankOperations {
    void addCustomer (Account a);
    int getNumOfCustomers ();
    Account getCustomer (String firstName);
    void void generateReport ();
}
```

?

\* Bank class :-

```
public class Bank implements BankOperations {
    private List<Account> AccountsList;
    static int NumOfCustomers;
    public Bank () { this.AccountsList = new
        ArrayList<> (); }
}
```

@ Override

```
public addCustomer (Account a) {
    AccountsList.add(a);
    NumOfCustomers++;
}
```

@override

```
public int getNumofCustomers() { return NumofCustomers; }
```

@Override

```
public Account getCustomer (String firstName) {
```

```
    for (Account a: AccountsList) {
```

```
        if (a.FirstName.equals (IgnoreCase (firstName)))
```

```
            return a;
```

```
    }  
    return null;
```

```
}
```

@override

```
public void generateReport () {
```

```
    System.out.println ("Customers Report \n" +  
        " " " ");
```

```
    for (Account a: AccountsList) {
```

```
        System.out.println ("Customer: " + a.FirstName + ", " + a.LastName  
            + " \n" + "Branch " + a.getBranch() + ", " +  
            a.branch.getServiceLevel());
```

```
        if (a.AccountType == 1)
```

```
            System.out.println ("Checking Account balance is " + a.balance);
```

```
        else if (a.AccountType == 2)
```

```
            System.out.println ("Saving Account balance is " + a.balance);
```

```
    }
```

```
}
```

```
}
```



→ Main class:—

```
public class main {
```

```
    public static void main (String[] args) {
```

```
        Bank bank = new Bank();
```

```
        Saving Account ashish = new SavingAccount  
            ("Ashish", "bhati", Branch.Bank);
```

```
        ashish.deposit(20000);
```

```
        bank.addCustomer(ashish);
```

```
        bank.addCustomer(new ChequingAccount
```

```
            ("Abhishek", "bhati", Branch.Bank);
```

```
        bank.generateReport();
```

```
        bank.getNumOfCustomers();
```

```
    }
```