Q. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if balance falls below this level a service charge imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the class Cur-acct and Sav-acct to make them more specific to their requirements. Include necessary methods to receive following tasks

a) Accept deposit customer and update the balance
b) Display the balance
c) Compute deposit interest
d) Permit the withdrawal and update the balance

Check for the minimum balance, penalty if necessary & update the balance

Ans:

```
import java.util.Scanner;

class Account{
    private String customerName;
    private String accountNumber;
    private String accountType;
    private double balance;
```

```
Account(String cname, String anum, String aType){
{
    customerName = ~~cname~~ cname;
    accountNumber = anum;
    accountType = aType;
    balance = 0.0;
}
public void deposit(double amount){
    if (amount > 0)
    {
        balance += amount;
        System.out.println("Deposited " + amount);
    }
    else
    {
        System.out.println("Invalid amount");
    }
}
public void displayBalance(){
    System.out.println("Balance : " + balance);
}

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

```
class SavAcct extends Account{
    private double interestrate;
    public SavAcct(String customerName, String actNumber,
                                    double interest){
        super(customerName, accountNumber, "Savings");
        this.interestRate = interest;
    }

    public void compoundInterest(int years){
        double balance = getBalance();
        double interest = balance * Math.pow((1 + interestrate/100),
                                    years) - balance;
        deposit(interest);
        System.out.println("Interest of $" + interest + " added");
    }

    public void withdraw(double amount){
        double balance = getbalance();
        if (amount > 0 && amount <= balance){
            updateBalance(balance - amount);
            System.out.println("Withdrawn : "+amount);
        }
        else {
            System.out.println("Insufficient Balance");
        }
    }
}
```

```
class CurrAcct extends Account)
{
    private double minBalance
    private double penalty;

    Publ
    CurrAcct (String customerName, String accNumber, double
                                    minBalance, double penalty)
    {
        super (customerName, accountNumber, "Current");
        this.minBalance = minBalance;
        this.penalty = penalty;
    }

public void withdraw(double account) {
        double balance = getBalance();
        if (amount > 0 && amount <= balance) {
            updateBalance (Balance - amount);
            System.out.println("Withdrawn : "+amount);
            if (getBalance() < minBalance) {
                updateBalance (get Balance() - penalty);
                System.out.println("Balance below minimum,
                                    Penalty of " + penalty);
            }
            else {
                System.out.println("Insufficient balance/
                                    invalid amount");
            }
        }
    }
```

```
public class Main
{ public static void main(String args[]) {
    Scanner Scanner = new Scanner(System.in);
    System.out.println("Enter name for Savings account");
    String savName = scanner.nextLine();
    System.out.printf("Enter the amount for Saving
                                    account :");
    String savAccNum = scanner.nextLine();
    SavAcct SavAccount = new SavAcct(savName, savAccNum,
                                    9.0);

    SavAccount.deposit(100);
    SavAccount.computeInterest(2);
    SavAcount.displayBalance();
    SavAccount.withdraw(500);
    System
    SavAccount.displayBalance();

    Sav
    System.out.printf("Enter customer name for Current
                                    Account :");

    String curName = Scanner.nextLine();
    System.out.printf("Enter account number for current
      account :");
    String curAccNum = scanner.nextLine();
    CurrAct curAccount = new CurAcct(curName, curAccNum,
                                    500,50);
```

```
        currAccount.deposit(1500);
        currAccount.displayBalance();
        currAccount.withdraw(1200);
        currAccount.displayBalance();
        currAccount.withdraw(500);
        currAccount.displayBalance();
    }
}
```

Output:

Enter customer name for Savings Account : John

Enter account number for Saving Account : 5123

Deposited : $1000.0

Interest of $81.599 added to your account

Balance : $1081.6

Withdrawn : $500.0

Balance : $581.6

Enter customer name for Current Account : Jane

Enter account number for Current Account : C456

Deposited : $1500.0

Balance : $1500

Withdrawn : $1200.0

Balance : $ 300

Balance below minimum . Penalty of $50.0 imposed

Balance : $ 250.0

Withdrawn : $500.0

Insufficient balance or invalid amount

Balance : $250.0