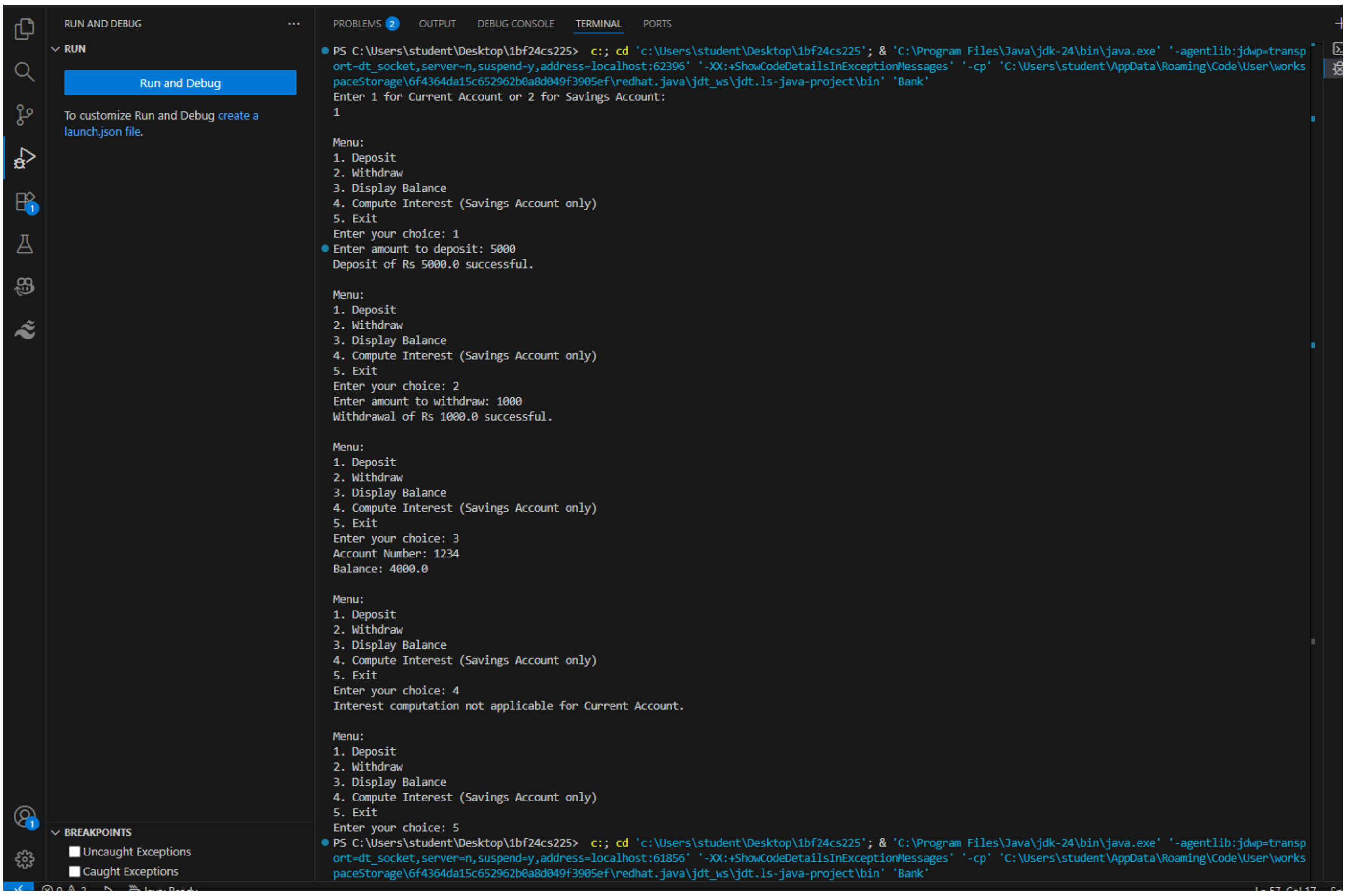
LAB 6:- Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides interest but no withdrawal facilities. The current account provides withdrawal facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a penalty is imposed. Create a class Account that stores customer name, account number. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```java
import java.util.Scanner;
class Account {
   String customerName;
   String accountNumber;
   double balance;

   Account(String customerName, String accountNumber) {
      this.customerName = customerName;
      this.accountNumber = accountNumber;
      this.balance = 0;
   }

   void deposit(double amount) {
      balance += amount;
      System.out.println("Deposit of Rs " + amount + " successful.");
   }

   void displayBalance() {
      System.out.println("Account Number: " + accountNumber + "\nBalance: " + balance);
   }
}

class SavingsAccount extends Account {
   SavingsAccount(String customerName, String accountNumber) {
      super(customerName, accountNumber);
   }

   void addInterest(double years) {
      double interestRate = 5;
      if (years <= 0) {
            System.out.println("No time passed, no interest added.");
            return;
        }
        double r = interestRate / 100.0;
        double interest = balance * r * years;
        balance += interest;
        System.out.println("Interest of Rs " + interest + " added.");
    }

    void withdraw(double amount) {
        System.out.println("Withdrawal not allowed for Savings Account.");
    }
}

class CurrentAccount extends Account {
    double minimumBalance = 1000;

    CurrentAccount(String customerName, String accountNumber) {
        super(customerName, accountNumber);
    }

    void withdraw(double amount) {
        if (balance - amount >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawal of Rs " + amount + " successful.");
        } else {
            System.out.println("Insufficient funds. Service charge applied.");
            imposePenalty();
        }
    }

    void imposePenalty() {
        double penalty = 200;
        balance -= penalty;
        System.out.println("Penalty of Rs " + penalty + " imposed.");
    }
}

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

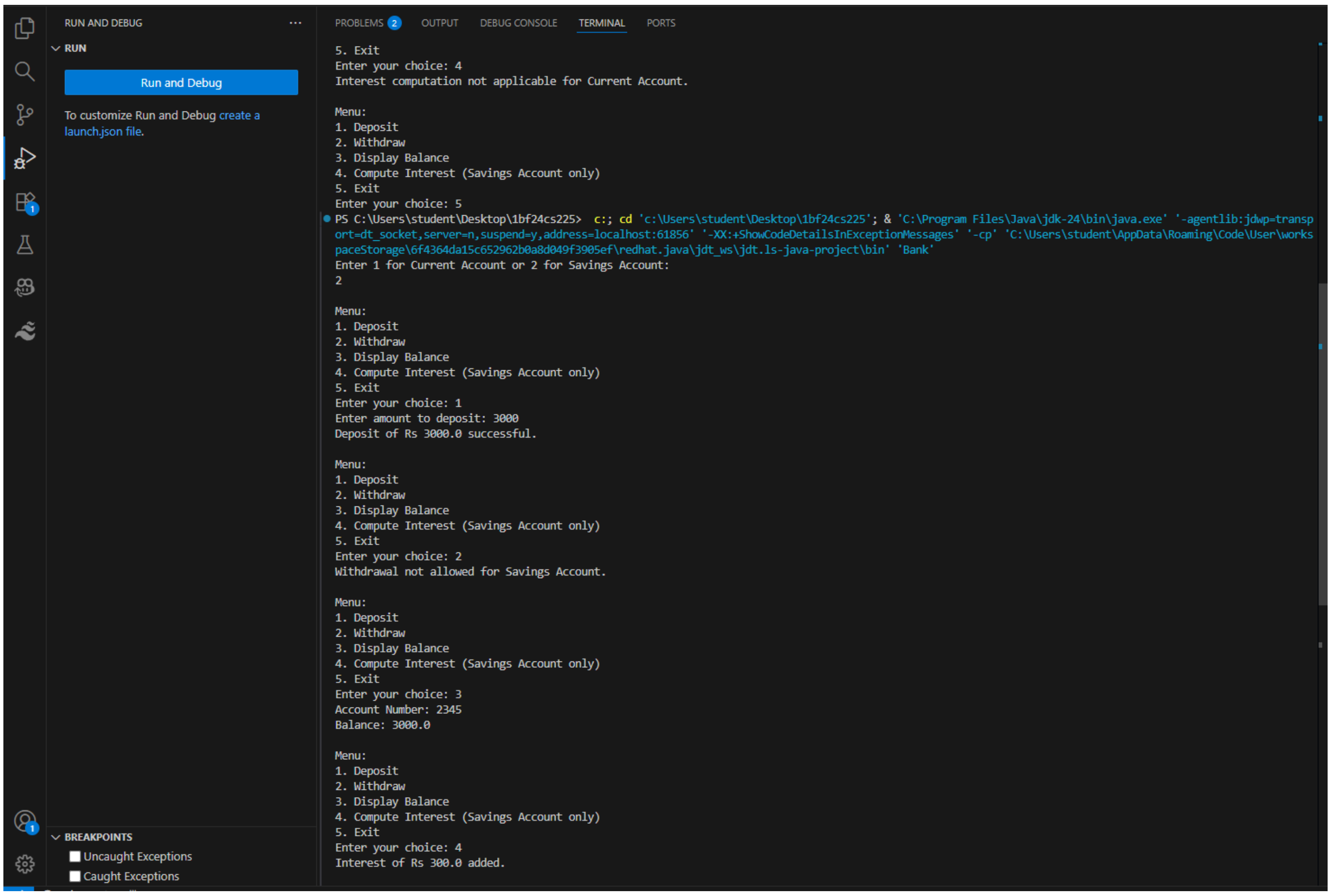
```
System.out.println("Enter 1 for Current Account or 2 for Savings Account:");
int choice1 = sc.nextInt();

Account acc;
if (choice1 == 1)
   acc = new CurrentAccount("Alice", "1234");
else
   acc = new SavingsAccount("James", "2345");

while (true) {
   System.out.println("\nMenu:");
   System.out.println("1. Deposit");
   System.out.println("2. Withdraw");
   System.out.println("3. Display Balance");
   System.out.println("4. Compute Interest (Savings Account only)");
   System.out.println("5. Exit");
   System.out.print("Enter your choice: ");
   int choice2 = sc.nextInt();

   switch (choice2) {
      case 1:
         System.out.print("Enter amount to deposit: ");
         double amount = sc.nextDouble();
         acc.deposit(amount);
         break;

      case 2:
         if (acc instanceof SavingsAccount) {
            ((SavingsAccount) acc).withdraw(0);
         } else {
            System.out.print("Enter amount to withdraw: ");
            amount = sc.nextDouble();
            ((CurrentAccount) acc).withdraw(amount);
         }
         break;

      case 3:
         acc.displayBalance();
         break;

      case 4:
```

```
                    if (acc instanceof SavingsAccount)
                        ((SavingsAccount) acc).addInterest(2);
                    else
                        System.out.println("Interest computation not applicable for Current
Account.");
                    break;

                case 5:
                    System.exit(0);
                    break;

                default:
                    System.out.println("Invalid choice.");
            }
        }
    }
}
```

OUTPUT:-



```
PS C:\Users\student\Desktop\1bf24cs225>  c:; cd 'c:\Users\student\Desktop\1bf24cs225'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-agentlib:jdwp=transp
ort=dt_socket,server=n,suspend=y,address=localhost:62396' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\student\AppData\Roaming\Code\User\works
paceStorage\6f4364da15c652962b0a8d049f3905ef\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'Bank'
Enter 1 for Current Account or 2 for Savings Account:
1

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 1
Enter amount to deposit: 5000
Deposit of Rs 5000.0 successful.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 2
Enter amount to withdraw: 1000
Withdrawal of Rs 1000.0 successful.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 3
Account Number: 1234
Balance: 4000.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 4
Interest computation not applicable for Current Account.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 5
PS C:\Users\student\Desktop\1bf24cs225>  c:; cd 'c:\Users\student\Desktop\1bf24cs225'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-agentlib:jdwp=transp
ort=dt_socket,server=n,suspend=y,address=localhost:61856' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\student\AppData\Roaming\Code\User\works
paceStorage\6f4364da15c652962b0a8d049f3905ef\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'Bank'
```



```
5. Exit
Enter your choice: 4
Interest computation not applicable for Current Account.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 5
PS C:\Users\student\Desktop\1bf24cs225>  c:; cd 'c:\Users\student\Desktop\1bf24cs225'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-agentlib:jdwp=transp
ort=dt_socket,server=n,suspend=y,address=localhost:61856' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\student\AppData\Roaming\Code\User\works
paceStorage\6f4364da15c652962b0a8d049f3905ef\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'Bank'
Enter 1 for Current Account or 2 for Savings Account:
2

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 1
Enter amount to deposit: 3000
Deposit of Rs 3000.0 successful.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 2
Withdrawal not allowed for Savings Account.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 3
Account Number: 2345
Balance: 3000.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 4
Interest of Rs 300.0 added.
```