Name:	**Debjit Ghosal**
UID:	**2023300065**
Experiment No.	**5**

AIM:	**Program on Inheritance**

Program 1

PROBLEM STATEMENT :

**The cost of stock on each day is given in an array A[] of size N. Day 1 price in first location, day 2 price in
second location etc. Find all the days on which you buy and sell the stock any number of times so that in
between those days your profit is maximum.A new transaction can only start after the previous
transaction is complete. Person can hold only one share at a time. Create a class Stock that has the name
of stock and array of prices. Also it has input method that initialises the predicted price of the stock in an
array of length N. Create class Transaction that is a sub class of Stock class. It has method
findMaximumProfit method.**

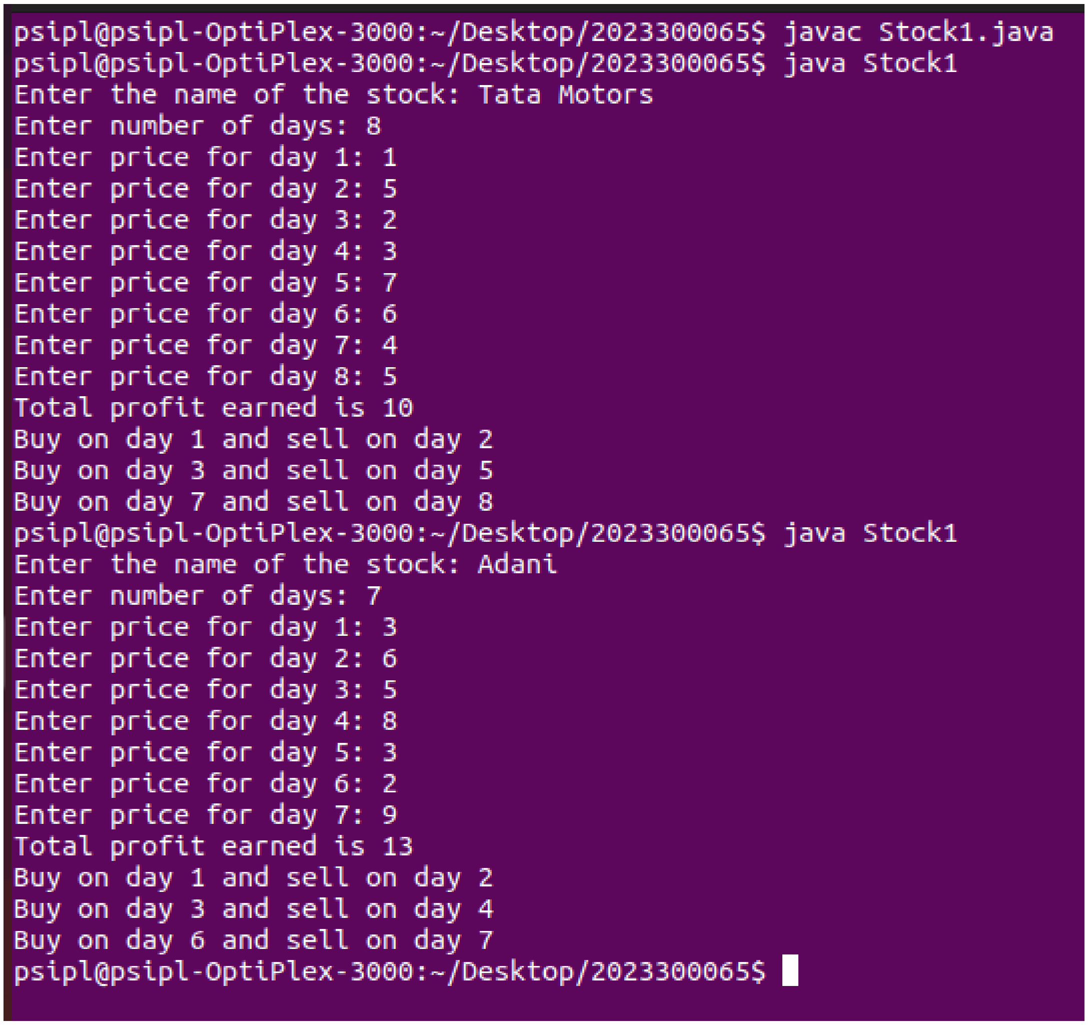
Example Stock Prices: {1, 5, 2, 3, 7, 6, 4, 5}
Total profit earned is 10
Buy on day 1 and sell on day 2
Buy on day 3 and sell on day 5
Buy on day 7 and sell on day 8

PROGRAM:

```
import java.util.*;

class Stock {
    String name;
    int[] prices;

    public Stock(String name, int[] prices) {
        this.name = name;
        this.prices = prices;
    }
```

```
}

class Transaction extends Stock {
    public Transaction(String name, int[] prices) {
        super(name, prices);
    }

    public void findMaximumProfit() {
        int maxProfit = 0;
        List<Integer> buyDays = new ArrayList<>();
        List<Integer> sellDays = new ArrayList<>();
        int buyDay = 0;
        int sellDay = 0;

        for (int i = 0; i < prices.length - 1; i++) {
            if (prices[i + 1] > prices[i]) {
                buyDays.add(i);
                buyDay = i;
                while (i < prices.length - 1 && prices[i + 1] >= prices[i]) {
                    i++;
                }
                sellDays.add(i);
                sellDay = i;
                maxProfit += prices[sellDay] - prices[buyDay];
            }
        }

        System.out.println("Total profit earned is " + maxProfit);
        for (int i = 0; i < buyDays.size(); i++) {
            System.out.println("Buy on day " + (buyDays.get(i) + 1) + " and
sell on day " + (sellDays.get(i) + 1));
        }
    }
}

public class Stock1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the name of the stock: ");
        String stockName = scanner.nextLine();

        System.out.print("Enter number of days: ");
        int n = scanner.nextInt();
        int[] prices = new int[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter price for day " + (i + 1) + ": ");
            prices[i] = scanner.nextInt();
        }

        Transaction transaction = new Transaction(stockName, prices);
        transaction.findMaximumProfit();
    }
}
```

RESULT:



```
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ javac Stock1.java
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ java Stock1
Enter the name of the stock: Tata Motors
Enter number of days: 8
Enter price for day 1: 1
Enter price for day 2: 5
Enter price for day 3: 2
Enter price for day 4: 3
Enter price for day 5: 7
Enter price for day 6: 6
Enter price for day 7: 4
Enter price for day 8: 5
Total profit earned is 10
Buy on day 1 and sell on day 2
Buy on day 3 and sell on day 5
Buy on day 7 and sell on day 8
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ java Stock1
Enter the name of the stock: Adani
Enter number of days: 7
Enter price for day 1: 3
Enter price for day 2: 6
Enter price for day 3: 5
Enter price for day 4: 8
Enter price for day 5: 3
Enter price for day 6: 2
Enter price for day 7: 9
Total profit earned is 13
Buy on day 1 and sell on day 2
Buy on day 3 and sell on day 4
Buy on day 6 and sell on day 7
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$
```

Program 2

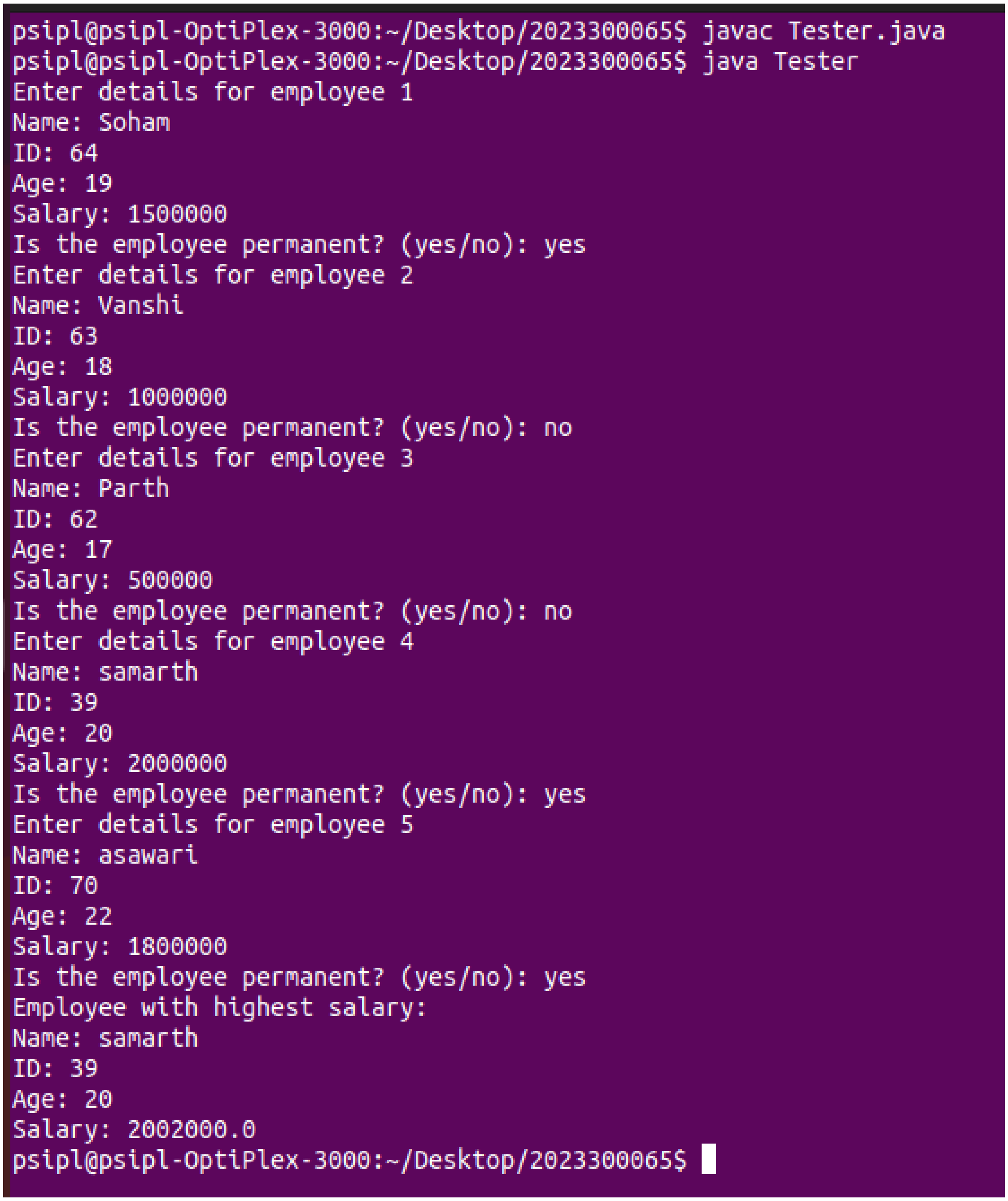
PROBLEM STATEMENT :

**Define parent class "Employee" that has 3 private attributes
String name, String id, int age.
Employee has constructor with 3 arguments that set value of name, id, age. It also has getter and setter
methods for all 3 private attributes.
Class "SalariedEmployee" is a sub class of Employee and has 1 private attribute empSalary.
"SalariedEmployee" can be permanent or on contract and has constructor SalariedEmployee(String name,
String id, int age, double empSalary) to set the values.
constructor SalariedEmployee must call the superclass constructor to set name, id, age and call setter
method to set the salary.
Employee salary is empSalary + 2000(allowance) if he is a permanent employee else Employee salary is
empSalary (no allowance).
Accept the details of 5 employees and print details of the employee with highest salary.
Create class Tester with main method**

PROGRAM:

```
import java.util.Scanner;

class Employee {
    private String name;
    private String id;
    private int age;

    public Employee(String name, String id, int age) {
        this.name = name;
        this.id = id;
        this.age = age;
    }

    // Getters and setters
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}

class SalariedEmployee extends Employee {
    private double empSalary;

    public SalariedEmployee(String name, String id, int age, double
empSalary) {
        super(name, id, age);
        this.empSalary = empSalary;
    }

    public double getEmpSalary() {
        return empSalary;
    }

    public void setEmpSalary(double empSalary) {
        this.empSalary = empSalary;
    }

    // Calculate salary with allowance for permanent employees
    public double calculateSalary() {
        return (this instanceof PermanentEmployee) ? empSalary + 2000 : empSalary;
    }
}

class PermanentEmployee extends SalariedEmployee {
    public PermanentEmployee(String name, String id, int age, double empSalary) {
        super(name, id, age, empSalary);
    }
}

class ContractEmployee extends SalariedEmployee {
    public ContractEmployee(String name, String id, int age, double empSalary) {
        super(name, id, age, empSalary);
    }
}

public class Tester {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        SalariedEmployee[] employees = new SalariedEmployee[5];

        for (int i = 0; i < 5; i++) {
            System.out.println("Enter details for employee " + (i + 1));
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("ID: ");
            String id = scanner.nextLine();
            System.out.print("Age: ");
            int age = scanner.nextInt();
            System.out.print("Salary: ");
            double salary = scanner.nextDouble();
            scanner.nextLine(); // Consume newline

            System.out.print("Is the employee permanent? (yes/no): ");
            String permanent = scanner.nextLine().toLowerCase();

            if (permanent.equals("yes")) {
                employees[i] = new PermanentEmployee(name, id, age,
salary);
            } else {
                employees[i] = new ContractEmployee(name, id, age, salary);
            }
        }

        // Finding employee with highest salary
        SalariedEmployee highestPaidEmployee = employees[0];
        for (int i = 1; i < employees.length; i++) {
            if (employees[i].calculateSalary() >
highestPaidEmployee.calculateSalary()) {
                highestPaidEmployee = employees[i];
            }
        }
                        System.out.println("  ");
        System.out.println("Employee with highest salary:");
        System.out.println("Name: " + highestPaidEmployee.getName());
        System.out.println("ID: " + highestPaidEmployee.getId());
        System.out.println("Age: " + highestPaidEmployee.getAge());
        System.out.println("Salary: " +
highestPaidEmployee.calculateSalary());
    }
}
```

RESULT:



```
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ javac Tester.java
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ java Tester
Enter details for employee 1
Name: Soham
ID: 64
Age: 19
Salary: 1500000
Is the employee permanent? (yes/no): yes
Enter details for employee 2
Name: Vanshi
ID: 63
Age: 18
Salary: 1000000
Is the employee permanent? (yes/no): no
Enter details for employee 3
Name: Parth
ID: 62
Age: 17
Salary: 500000
Is the employee permanent? (yes/no): no
Enter details for employee 4
Name: samarth
ID: 39
Age: 20
Salary: 2000000
Is the employee permanent? (yes/no): yes
Enter details for employee 5
Name: asawari
ID: 70
Age: 22
Salary: 1800000
Is the employee permanent? (yes/no): yes
Employee with highest salary:
Name: samarth
ID: 39
Age: 20
Salary: 2002000.0
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ 
```

CONCLUSION:	I have learnt about inheritance.I have learnt how to use extend parent class properties to child class. I have also learnt that we need to use super keyword for calling a parametrized default class.