Name	**Hatim Yusuf Sawai**
UID no.	**2021300108**
Experiment No.	**5**

AIM:	**Program on Inheritance: Implement a Program to demonstrate single, multilevel Inheritance**

Program 1

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

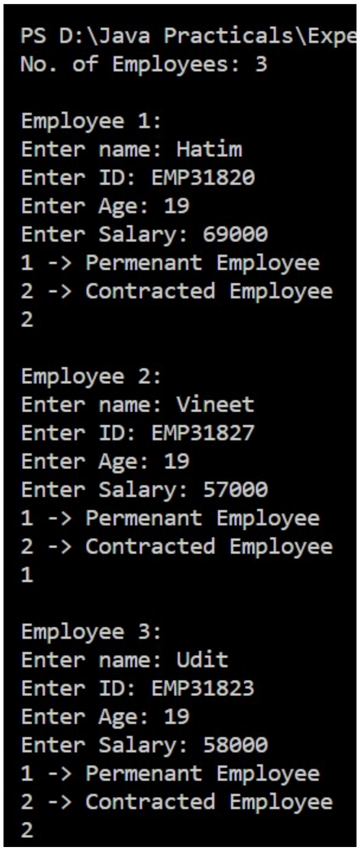
PROGRAM:

```
import java.util.*;
class Employee {
    private int age;
    private String name,id;
    Employee(String name,String id,int age) {
        this.name = name;
        this.id = id;
        this.age = age;
    }
    //getters
    int getage(){return age;}
    String getid(){return id;}
    String getname(){return name;}
    //setters
    void setid(String id){this.id = id;}
    void setname(String name){this.name = name;}
    void setage(int age){this.age = age;}
}
class SalariedEmployee extends Employee {
    private double salary;
    SalariedEmployee(String name,String id,int age,double empSalary) {
        super(name,id,age);
        setSalary(empSalary);
    }
    //setters & getters
    void setSalary(double salary){this.salary = salary;}
    double getSalary(){return salary;}
    void sortsal(int n,SalariedEmployee [] employees) {
        double max = 0;
        int c=0;
        for(int i=0;i<n;i++) {
            if(employees[i].getSalary()>max) {
                max = employees[i].getSalary();
                c = i;
            }
        }
        System.out.println("\nDetails of Highest-Paid
Employee:\nEmployee ID\tName\tAge\tSalary");

System.out.printf("%s\t%s\t%d\t%.2f",employees[c].getid(),employee
s[c].getname(),employees[c].getage(),employees[c].getSalary());
    }
}
public class Tester {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String name,id;
        int age,e;
        double salary;
        System.out.print("No. of Employees: ");
        int n = sc.nextInt();
        SalariedEmployee [] employees = new SalariedEmployee[n];
        for(int i=0;i<n;i++) {
            System.out.printf("\nEmployee %d:\n",i+1);
            System.out.print("Enter name: ");
            name = sc.next();
            System.out.print("Enter ID: ");
            id = sc.next();
            System.out.print("Enter Age: ");
            age = sc.nextInt();
            System.out.print("Enter Salary: ");
            salary = sc.nextDouble();
            System.out.print("1 -> Permenant Employee\n2 -> Contracted
Employee\n");
            e = sc.nextInt();
            if(e==1) {
                salary += 2000;
            }
            employees[i] = new SalariedEmployee(name,id,age,salary);
        }
        employees[0].sortsal(n,employees);
        sc.close();
    }
}
```

RESULT:

Details:



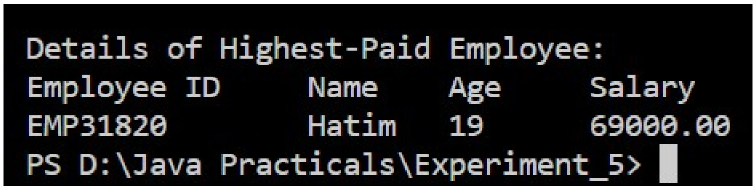
```
PS D:\Java Practicals\Expe
No. of Employees: 3

Employee 1:
Enter name: Hatim
Enter ID: EMP31820
Enter Age: 19
Enter Salary: 69000
1 -> Permenant Employee
2 -> Contracted Employee
2

Employee 2:
Enter name: Vineet
Enter ID: EMP31827
Enter Age: 19
Enter Salary: 57000
1 -> Permenant Employee
2 -> Contracted Employee
1

Employee 3:
Enter name: Udit
Enter ID: EMP31823
Enter Age: 19
Enter Salary: 58000
1 -> Permenant Employee
2 -> Contracted Employee
2
```

Result:



```
Details of Highest-Paid Employee:
Employee ID     Name     Age     Salary
EMP31820        Hatim    19      69000.00
PS D:\Java Practicals\Experiment_5>
```

Program 2

PROBLEM STATEMENT:	Mr. Abhishek Bachan is an H.R. Manager of “ABCL Technologies” At the beginning of the new year he anticipated that the company will need 30 new additional persons to fill up different vacancies in software team. He gave an advertisement in the newspaper inviting applications for

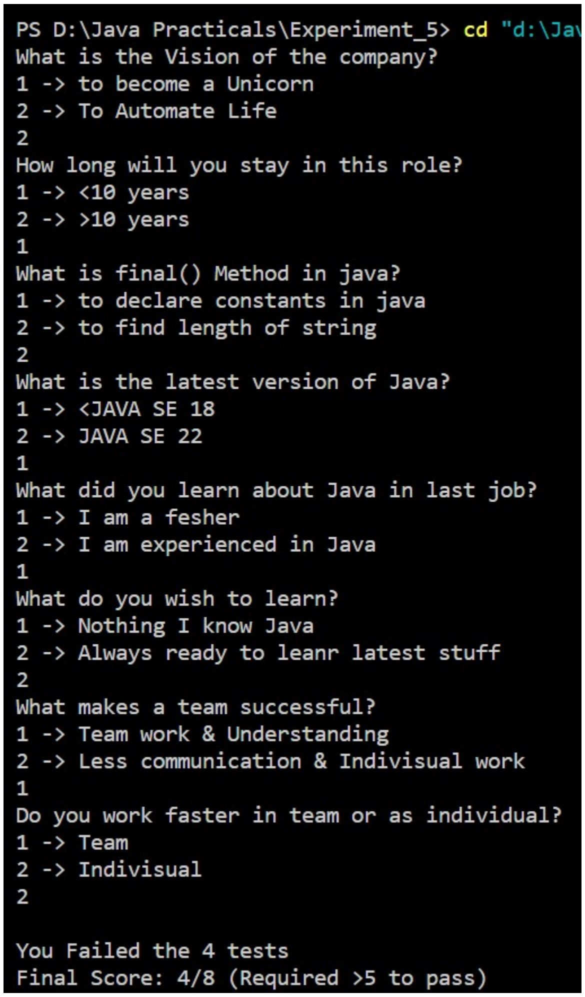
	filling up different vacant posts. As many as 120 applications were received. The same were scrutinized and they were given four tests. There are 4 sections and each section has 2 questions each. Create class Testing that generates the marks based on his answers in test. class Testing has methods takeTest. Design objective answers for following questions and conduct test and generate result whether candidate passed or failed. Create subclass Recruitment that prints if the person has passed(greater than 60%) or failed the test. generateResult() generates the marks percentage based on his answers in test. Create class TesterAB with main method The objective of the first test was to find out how much interest the applicant takes in his work. Questions asked 1.What is the vision of the company? 2.How long will you stay in this role? The objective of the second test was to find out 'specialization' of the applicant in any particular area. Questions asked 1.What is final method in Java? 2.What is the latest version of Java? Third test aimed at making sure whether the applicant was capable of learning through training or not. Questions asked 1.What did you learn about Java in last job? 2.What do you wish to learn? The purpose of the fourth test was to find out how much capability a person has to mix-up with other persons, and whether he can influence other persons and get influenced by them. Questions asked 1.What makes a team successful? 2.Do you work faster in team or as individual?

PROGRAM:

```java
import java.util.*;
class Testing {
    int score,x;
    int[] ans = new int[8];
    int[] correct = {2,2,1,1,2,2,1,1};
    Scanner sc = new Scanner(System.in);
    void takeTest() {
        System.out.println("What is the Vision of the company?\n1 -> to become a Unicorn\n2 -> To Automate Life");
        ans[0] = sc.nextInt();
        System.out.println("How long will you stay in this role?\n1 -> <10 years\n2 -> >10 years");
        ans[1] = sc.nextInt();
        System.out.println("What is final() Method in java?\n1 -> to declare constants in java\n2 -> to find length of string");
        ans[2] = sc.nextInt();
        System.out.println("What is the latest version of Java?\n1 -> <JAVA SE 18\n2 -> JAVA SE 22");
        ans[3] = sc.nextInt();
        System.out.println("What did you learn about Java in last job?\n1 -> I am a fesher\n2 -> I am experienced in Java");
        ans[4] = sc.nextInt();
        System.out.println("What do you wish to learn?\n1 -> Nothing I know Java\n2 -> Always ready to leanr latest stuff");
        ans[5] = sc.nextInt();
        System.out.println("What makes a team successful?\n1 -> Team work & Understanding\n2 -> Less communication & Indivisual work");
        ans[6] = sc.nextInt();
        System.out.println("Do you work faster in team or as individual?\n1 -> Team\n2 -> Indivisual");
        ans[7] = sc.nextInt();
    }
}
class Recruitment extends Testing {
    Scanner sc = new Scanner(System.in);
    void generateResult() {
        for(int i=0;i<8;i++) {
            if(correct[i] == ans[i]) {
                score += 1;
            }
        }
        if(score>=5) {
            System.out.printf("\nYou passed the 4 tests\nFinal Score:
%d/8\n",score);
        }
        else {
            System.out.printf("\nYou Failed the 4 tests\nFinal Score:
%d/8 (Required >5 to pass)\n",score);
        }
    }
    void sortapps(Recruitment [] recruits) {
        for(int i=0;i<recruits.length-1;i++) {
            for(int j=i+1;j<recruits.length;j++) {
                if(recruits[i].score<recruits[j].score) {
                    Recruitment temp = recruits[i];
                    recruits[i] = recruits[j];
                    recruits[j] = temp;
                }
            }
        }
        System.out.println("The 3 Applicants who got Selected
are:\nApplicant No.\tScore(%)");
        for(int i=0;i<3;i++) {
            System.out.printf("Applicant
%d\t%.1f%\n",recruits[i].x,((double)(recruits[i].score)/8)*100);
        }
    }
}
public class TesterAB {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Recruitment[] recruits = new Recruitment[6];
        for(int i=0;i<6;i++) {
            recruits[i] = new Recruitment();
            recruits[i].x = i+1;
            recruits[i].takeTest();
            recruits[i].generateResult();
        }
        recruits[0].sortapps(recruits);
        sc.close();
    }
}
```

RESULT:

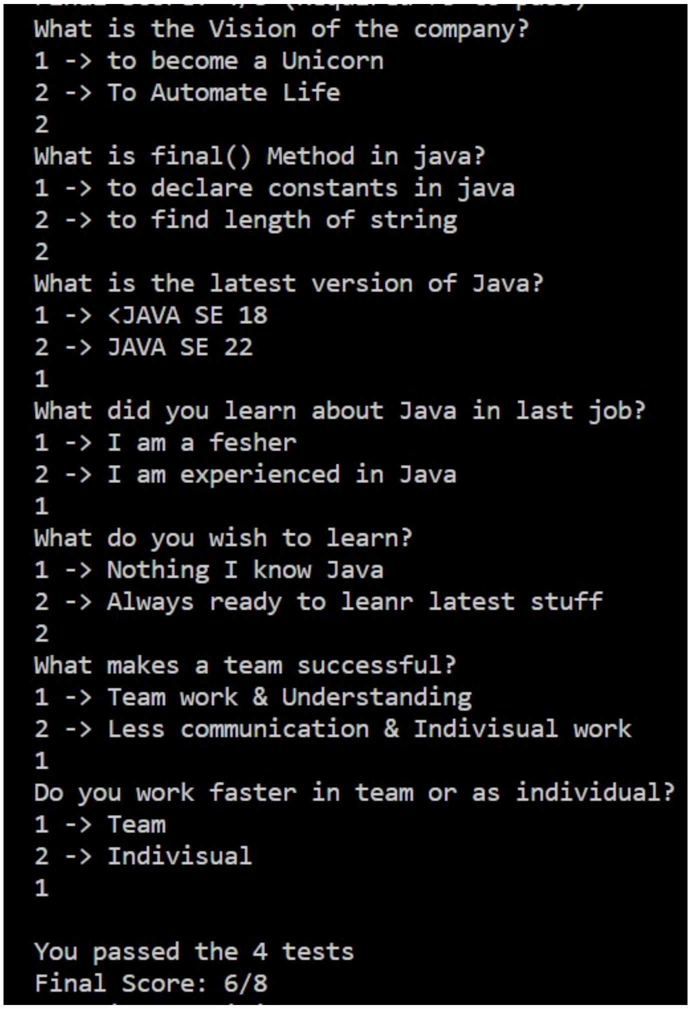
Fail Case:



```
PS D:\Java Practicals\Experiment_5> cd "d:\Ja
What is the Vision of the company?
1 -> to become a Unicorn
2 -> To Automate Life
2
How long will you stay in this role?
1 -> <10 years
2 -> >10 years
1
What is final() Method in java?
1 -> to declare constants in java
2 -> to find length of string
2
What is the latest version of Java?
1 -> <JAVA SE 18
2 -> JAVA SE 22
1
What did you learn about Java in last job?
1 -> I am a fesher
2 -> I am experienced in Java
1
What do you wish to learn?
1 -> Nothing I know Java
2 -> Always ready to leanr latest stuff
2
What makes a team successful?
1 -> Team work & Understanding
2 -> Less communication & Indivisual work
1
Do you work faster in team or as individual?
1 -> Team
2 -> Indivisual
2

You Failed the 4 tests
Final Score: 4/8 (Required >5 to pass)
```

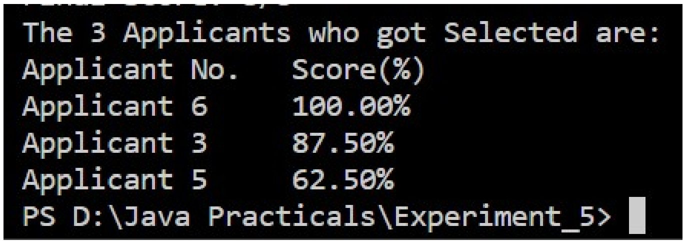
Pass Case:



```
What is the Vision of the company?
1 -> to become a Unicorn
2 -> To Automate Life
2
What is final() Method in java?
1 -> to declare constants in java
2 -> to find length of string
2
What is the latest version of Java?
1 -> <JAVA SE 18
2 -> JAVA SE 22
1
What did you learn about Java in last job?
1 -> I am a fesher
2 -> I am experienced in Java
1
What do you wish to learn?
1 -> Nothing I know Java
2 -> Always ready to leanr latest stuff
2
What makes a team successful?
1 -> Team work & Understanding
2 -> Less communication & Indivisual work
1
Do you work faster in team or as individual?
1 -> Team
2 -> Indivisual
1

You passed the 4 tests
Final Score: 6/8
```

Result:



```
The 3 Applicants who got Selected are:
Applicant No.   Score(%)
Applicant 6     100.00%
Applicant 3     87.50%
Applicant 5     62.50%
PS D:\Java Practicals\Experiment_5> 
```

Program 3

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 time so that in between those days your profit is maximum.A new transaction can only start after previous transaction is complete. Person can hold

only one share at a time.

Create class **Stock** that has 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 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

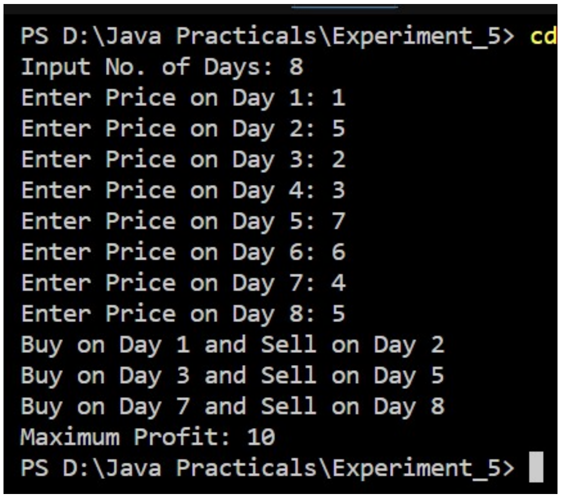
Buy on day 3 and sell on day 5

Buy on day 7 and sell on day 8

PROGRAM:

```
import java.util.*;
class Stock {
    Scanner sc = new Scanner(System.in);
    int[] prices;
    int n;
    void input() {
        System.out.print("Input No. of Days: ");
        n = sc.nextInt();
        prices = new int[n];
        for(int i=0;i<n;i++) {
            System.out.printf("Enter Price on Day %d: ",i+1);
            prices[i] = sc.nextInt();
        }
    }
}
class transaction extends Stock {
    void findMaxProfit() {
        int profit=0,diff=0,k=0;
        for(int i=0;i<n;i++) {
            diff = 0;
            for(int j=i+1;j<n;j++) {
                if(diff<prices[j]-prices[i]) {
                    diff = prices[j]-prices[i];
                    k=j;
                }
                else if(diff>=prices[j]-prices[i]) {
                    break;
                }
            }
            if(diff!=0) {
                profit += diff;
                System.out.printf("Buy on Day %d and Sell on Day 
%d\n",i+1,k+1);
                i=k;
            }
        }
        System.out.println("Maximum Profit: "+profit);
    }
}
public class TesterST {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        transaction t1 = new transaction();
        t1.input();
        t1.findMaxProfit();
        sc.close();
    }
}
```

RESULT:



```
PS D:\Java Practicals\Experiment_5> cd
Input No. of Days: 8
Enter Price on Day 1: 1
Enter Price on Day 2: 5
Enter Price on Day 3: 2
Enter Price on Day 4: 3
Enter Price on Day 5: 7
Enter Price on Day 6: 6
Enter Price on Day 7: 4
Enter Price on Day 8: 5
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
Maximum Profit: 10
PS D:\Java Practicals\Experiment_5>
```

CONCLUSION:	In this experiment, we learnt inheritance in java, use of super() method to declare constructors of subclasses and how to write complex programs using inherited classes.