Name: Sahil Hedau

Sec: A (A3)

Roll No.: 56

Date: 13/5/2023

**OOPs Practical 2**

**Aim:** Create an Employee Attendance Management System (EMS).  
A. Design a Class Time with data members as hours, minutes and seconds. This class provides the functionality to Add and Subtract two time objects. The result will be returned in a new time object.  
B. Create a class to store Attendance. The attendance will have Time\_IN , Time\_OUT along with date information.  
C. Create a class Employee with employee name,  id, name, age, designation, salary, Attendance[].   
a. Provide the functionality to count the number of hours worked in a month.  
b. Display the number of working hours of an employee for a given date.  
Write appropriate constructors and display function where required.

**Code & Output:**

**time.java**

public class time {

    int hrs,min,sec;

    time(int hrs,int min, int sec){

        this.hrs = hrs;

        this.min = min;

        this.sec = sec;

    }

    time addTime(*time* a, *time* b){

        time c = new time(0,0,0);

        c.hrs = a.hrs+b.hrs;

        int temp\_min = a.min+b.min;

        if(temp\_min >= 60){

            c.hrs += 1;

            temp\_min -= 60;

        }

        c.min = temp\_min;

        int temp\_sec = a.sec+b.sec;

        if(temp\_sec >= 60){

            c.min += 1;

            temp\_sec -= 60;

        }

        c.sec = temp\_sec;

        return c;

    }

    void display(*time* a){

        System.out.print(a.hrs + " hrs " + a.min + " min " + a.sec + " sec ");

    }

    time subTime(*time* a, *time* b){

        time max = new time(0,0,0);

        time mini = new time(0,0,0);

        time ans = new time(0,0,0);

        if(a.hrs < b.hrs){

            max = b;

            mini = a;

        }

        else{

            max = a;

            mini = b;

        }

        ans.hrs = max.hrs - mini.hrs;

        if(max.min >= mini.min){

            ans.min = max.min - mini.min;

        }

        else{

            ans.hrs--;

            ans.min = 60 - mini.min;

        }

        if(max.sec >= mini.sec){

            ans.sec = max.sec - mini.sec;

        }

        else{

            ans.min--;

            ans.sec = 60 - max.sec;

        }

        return ans;

    }

}

**attend.java**

public class attend {

    time time\_in = new time(0,0,0);

    time time\_out = new time(0,0,0);

    int date, month, year;

    attend(*time* time\_in, *time* time\_out, int date, int month, int year){

        this.time\_in = time\_in;

        this.time\_out = time\_out;

        this.date = date;

        this.month = month;

        this.year = year;

    }

    void display(){

        System.out.print("\nDate: " + date +" "+ month+" " + year);

        System.out.print("\nTime in: ");

        time\_in.display(time\_in);

        System.out.print("\nTime out: ");

        time\_out.display(time\_out);

    }

}

**employee.java**

public class employee {

    String name;

    int id;

    int age;

    String des;

    int sal;

    attend[] a = new attend[10];          // Max days is set to 10 for now.

    int days=0;

    private time t = new time(0,0, 0);

    employee(String name, int id, int age, String des, int sal){

        this.name = name;

        this.id = id;

        this.age = age;

        this.des = des;

        this.sal = sal;

    }

    void addAttend(int date, int month, int year, *time* tin, *time* tout){

        a[days] = new attend(tin, tout, date, month, year);

        days++;

    }

    void display(){

        System.out.println("\nName: "+name);

        System.out.println("ID: "+id);

        System.out.println("Age: "+age);

        System.out.println("Designation: "+des);

        System.out.println("Salary: "+sal);

    }

    time workingHourInMonth(int m){

        time hr = new time(0,0,0);

        for(int i=0;i<days;i++){

            if(a[i].month == m){

                time temp = t.subTime(a[i].time\_in, a[i].time\_out);

                hr = t.addTime(hr, temp);

            }

        }

        return hr;

    }

    int displayWorkingHoursOnDate(int date, int month, int year){

        for(int i=0;i<days;i++){

            if(date == a[i].date && month == a[i].month && year == a[i].year){

                time tt = t.subTime(a[i].time\_in, a[i].time\_out);

                System.out.println("\n\nWorking Hours On Date: "+date+" "+month+" "+year);

                System.out.print("Working Time: ");

                tt.display(tt);

                return 1;

            }

        }

        return 0;

    }

}

**main.java**

public class main {

    public static void main(String[] args) {

        // Part A

        System.out.println("\n----------- Part A -----------");

        time t1 = new time(10,10,10);

        time t2 = new time(17,5,5);

        System.out.print("\nTime 1: ");

        t1.display(t1);

        System.out.print("\nTime 2: ");

        t2.display(t2);

        time s = new time(0,0,0);

        System.out.print("\n\nAddition of time: ");

        s = s.addTime(t1, t2);

        s.display(s);

        time s1 = new time(0,0,0);

        System.out.print("\nSubtraction of time: ");

        s1 = s1.subTime(t1, t2);

        s1.display(s1);

        // Part B

        System.out.println("\n\n----------- Part B -----------");

        attend a = new attend(t1, t2, 30, 12, 2003);

        a.display();

        // Part C

        System.out.println("\n\n----------- Part C -----------");

        employee[] ea = new employee[5];

        time timein = new time(10, 30, 13);

        time timeout = new time(17, 0, 24);

        ea[0] = new employee("Sahil", 56, 19, "SDE", 50000);

        ea[0].addAttend(1,1,23,timein,timeout);

        ea[0].addAttend(2,1,23,timein,timeout);

        ea[0].addAttend(3,1,23,timein,timeout);

        ea[0].addAttend(4,1,23,timein,timeout);

        ea[0].addAttend(5,1,23,timein,timeout);

        ea[0].display();

        time temp = ea[0].workingHourInMonth(1);

        System.out.print("\nWorking Hours in month 1: ");

        temp.display(temp);

        ea[0].displayWorkingHoursOnDate(2,1,23);

    }

}

**Output:**

