

CO24557: Object Oriented Programming

Assignment 3

Submitted by: Sarvgya Sharma

0801CS161079

Signature of Professor

BE-II Year (CSE)

```
//CO24557 Assignment 3 - problem 1
//class Date : sets and shows date given by user
//class DateTest : takes date as input from user and implements the
//Date class

import java.util.Scanner;

class Date
{
    private int year,month,day ;

    public Date(int year,int month,int day)
    {
        this.year = year;
        this.month = month;
        this.day = day;
    }

    public void setYear(int year)
    {    this.year = year;    }

    public void setMonth(int month)
    {    this.month = month;    }

    public void setDay(int day)
    {    this.day = day;    }

    public int getYear()
    { return year;    }

    public int getMonth()
    { return month;    }

    public int getDay()
    { return day;    }

    public void displayDate()
    {
```

```

        System.out.print("\nDate: "+month+"/"+day+"/"+year);
    }

}

public class DateTest
{
    public static void main(String[] args)
    {
        int year,month,day;
        Scanner stdin = new Scanner(System.in);

        System.out.print("\nEnter Year: ");
        year = stdin.nextInt();

        System.out.print("\nEnter Month: ");
        month = stdin.nextInt();
        while(month>12||month<=0)
        {
            System.out.print("Invalid Month!\nEnter month again:
");
            month = stdin.nextInt();
        }

        System.out.print("\nEnter day: ");
        day = stdin.nextInt();
        while(day>31||day<=0)
        {
            System.out.print("Invalid Day!\nEnter day again: ");
            day = stdin.nextInt();
        }

        Date date = new Date(year,month,day);
        date.displayDate();

        System.out.print("\n\nDo you want to change this
date(y/n)? ");
        char ch = stdin.next().charAt(0);
        if(ch=='y' || ch=='Y')
        {
            System.out.print("\nEnter Year: ");
            year = stdin.nextInt();
            date.setYear(year);

            System.out.print("\nEnter Month: ");
            month = stdin.nextInt();
            while(month>12||month<=0)
            {
                System.out.print("\nInvalid Month!\nEnter
month again: ");
                month = stdin.nextInt();
            }
            date.setMonth(month);
        }
    }
}

```

```

        System.out.print("\nEnter day: ");
        day = stdin.nextInt();
        while(day>31||day<=0)
        {
            System.out.print("\nInvalid Day!\nEnter day
again: ");
            day = stdin.nextInt();
        }
        date.setDay(day);
        System.out.print("\nDate changed succesfully! ");
        date.displayDate();
    }
}

```

```

//CO 24557 : Assignment 3 - problem 2
//super class : Car ; sub-classes : Truck,Sedan

```

```

class Car
{
    int speed;
    double regularPrice;
    String color;

    public Car() {}

    public Car(int speed,double regularPrice,String color)
    {
        this.speed = speed;
        this.regularPrice = regularPrice;
        this.color = color;
    }

    double getSalePrice()
    { return regularPrice; }
}

class Truck extends Car
{
    int weight;

    public Truck() {}

    public Truck(int weight,int speed,double regularPrice,String
color)
    {
        super(speed,regularPrice,color);
        this.weight = weight;
    }

    double getSalePrice()
    {
        if(weight>2000)           //10% discount

```

```

        return super.getSalePrice() -
(0.1*super.getSalePrice());

        else //20% discount
            return super.getSalePrice() -
(0.2*super.getSalePrice());
    }
}

class Ford extends Car
{
    int year;
    int manufacturerDiscount;

    public Ford() {}

    public Ford(int year,int discount,int speed,double
regularPrice,String color)
    {
        super(speed,regularPrice,color);
        this.year = year;
        manufacturerDiscount = discount;
    }

    double getSalePrice()
    {
        return super.getSalePrice()-manufacturerDiscount;
    }
}

class Sedan extends Car
{
    int length;

    public Sedan() {}

    public Sedan(int length,int speed,double regularPrice,String
color)
    {
        super(speed,regularPrice,color);
        this.length = length;
    }

    double getSalePrice()
    {
        if(length>20) //5% discount
            return super.getSalePrice() -
(0.05*super.getSalePrice());

        else //10% discount
            return super.getSalePrice() -
(0.1*super.getSalePrice());
    }
}

public class MyOwnAutoShop

```

```

{
    public static void main(String[] args)
    {
        Sedan car1 = new Sedan(25,230,1250000.0,"Natural
silver");
                                //feet,kmph,Rs.,color

        Ford car2 = new Ford(2016,50000,180,510000.0,"Pearl
White");
                                //year,Rs.(discount),kmph,Rs.,color

        Ford car3 = new Ford(2018,20000,200,630000.0,"Ocean
Blue");
                                //year,Rs.(discount),kmph,Rs.,color

        System.out.print("\n\nCars available in the Auto shop
are:");

        System.out.print("\n\nCar 1:");
        System.out.print("\nType: Sedan");
        System.out.print("\nLength: "+car1.length+" feet");
        System.out.print("\nSpeed: "+car1.speed+" Kmph");
        System.out.print("\nColor: "+car1.color);
        System.out.print("\nRegular Price:
Rs."+car1.regularPrice);
        System.out.print("\nSale Price:
Rs."+car1.getSalePrice());

        System.out.print("\n\nCar 2:");
        System.out.print("\nType: Ford");
        System.out.print("\nYear: "+car2.year);
        System.out.print("\nSpeed: "+car2.speed+" Kmph");
        System.out.print("\nColor: "+car2.color);
        System.out.print("\nRegular Price:
Rs."+car2.regularPrice);
        System.out.print("\nManufacturer discount:
Rs."+car2.manufacturerDiscount);
        System.out.print("\nSale Price:
Rs."+car2.getSalePrice());

        System.out.print("\n\nCar 3:");
        System.out.print("\nType: Ford");
        System.out.print("\nYear: "+car3.year);
        System.out.print("\nSpeed: "+car3.speed+" Kmph");
        System.out.print("\nColor: "+car3.color);
        System.out.print("\nRegular Price:
Rs."+car3.regularPrice);
        System.out.print("\nManufacturer discount:
Rs."+car3.manufacturerDiscount);
        System.out.print("\nSale Price:
Rs."+car3.getSalePrice());

    }
}

```

```

//CO24557 Assignment 3 - problem 3
//Guess the number game

import java.util.*;

class GuessTheNumber
{
    public static void main(String[] args)
    {
        char ch = 'y' ;
        while(ch=='y' || ch=='Y')
        {
            System.out.println("\n GUESS THE NUMBER GAME (1 to
1000) ");

            System.out.print("\n1.Easy \n2.Normal \n3.Hard");
            System.out.print("\n\n Select the difficulty level:
");

            Scanner stdin = new Scanner(System.in);
            int level = stdin.nextInt();
            int chances = 9;
            switch(level)
            {
                case 1: chances = 13;
                    break;

                case 2: chances = 10;
                    break;

                case 3: chances = 7;
                    break;

                default: System.out.println("\n Wrong
choice!");
                    main(new String[1]);
            }

            Random rand = new Random();
            int key = rand.nextInt(1000)+1;
            int i;
            for(i=chances;i>0;--i)
            {
                System.out.printf("\n Enter your Guess(%d
chances left): ",i);

                int guess = stdin.nextInt();
                if(guess == key)
                {
                    System.out.print("\n\n Congratulations!
You guessed the number.");
                    break;
                }
                if(guess<key)
                    System.out.println(" Too low. Try
again.");
                else

```

```

        System.out.println(" Too High. Try
again.");
    }

    if(i==0)
        System.out.print("\n Sorry you lost. The
number was "+key);

    System.out.print("\n\n Do you want to play
again(y/n)? ");
    ch = stdin.next().charAt(0);

    }
}

```

```

//CO24557 Assignment 3 - problem 4
//simple bank account activities;2 types of account - saving and
//current

import java.util.*;

abstract class Account
{
    String CustomerName;
    private int AccountNumber;
    private double Balance;
    String AccountType;

    public Account() {}

    public Account(String name,String type,double money)
    {
        CustomerName = name;
        AccountType = type;;
        Balance = money;
        Random rand = new Random();
        AccountNumber = rand.nextInt(10000000)+1000000; //random
number between 1000000 to 9999999
    }

    public double getBalance()
    {    return Balance; }

    public void setBalance(double money)
    {    Balance = money; }

    public void deposit(double money,int time)
    {
        this.calculateInterest(time);
        Balance = Balance + money;
        System.out.print("\n\nRs."+money+" deposited
successfully.");
    }
}

```

```

    }

    public void withdraw(double money,int time)
    {
        this.calculateInterest(time);

        if(money>this.getBalance() || this.getBalance()<0)
            System.out.print("\nCannot withdraw more than your
balance in account!");
        else
        {
            Balance = Balance - money;
            System.out.print("\n\nRs."+money+" withdrwan
successfully.");
        }
    }

    public abstract void calculateInterest(int time);

    public void showDetails()
    {
        System.out.print("\n\nMini Statement:");
        System.out.print("\nCustomer Name: "+CustomerName);
        System.out.print("\nAccount Type: "+AccountType);
        System.out.print("\nAccount Number: "+AccountNumber);
        System.out.print("\nAccount Balance: Rs."+Balance);
    }
}

class SavingsAccount extends Account
{
    public SavingsAccount(String name,String type,double money)
    {
        super(name,type,money);
    }

    public void calculateInterest(int time)           //compound
interest
    {
        double interest;

        while(time>0)
        {
            interest = this.getBalance()*0.005;           //0.5%
interest rate per month
            this.setBalance(this.getBalance()+interest);
            time--;
        }
    }
}

class CurrentAccount extends Account
{
    public CurrentAccount(String name,String type,double money)
    {

```



```

        super(name,type,money);
    }

    public void calculateInterest(int time)           //simple
interest
    {
        double interest = (this.getBalance()*0.003*time)/100;
        //0.3% interest rate per month
        this.setBalance(this.getBalance()+interest);
    }

    public void withdraw(double money, int time)
    //withdraw func overridden
    {
        this.calculateInterest(time);

        if(money>this.getBalance() || this.getBalance()<0)
            System.out.print("\nCannot withdraw more than your
balance in account!");
        else
        {
            this.setBalance(this.getBalance() - money);
            System.out.print("\n\nRs."+money+" withdrwan
successfully.");
            if(this.getBalance()<1000)
            {
                System.out.print("\nBalance in account is less
than RS.1000(min. limit)!\nImposing penalty of Rs.500");
                this.setBalance(this.getBalance() - 500);
            }
        }
    }
}

public class AccountTester
{
    public static void main(String[] args)
    {
        String name;
        double money = 0;
        int time;
        int choice = 0;
        Scanner input = new Scanner(System.in);

        System.out.print("\n\n ----Banking System----");
        System.out.print("\n\nSelect Account type: \n1.Current
Account \n2.Savings Account");
        System.out.print("\n\nChoice: ");
        int type = input.nextInt();
        input.nextLine();           //flushing buffer

        switch(type)
        {
            case 1: System.out.print("\n\nEnter Customer Name:
");

```

```

        name = input.nextLine();
        int flag=0;
        while(flag==0)
        {
            System.out.print("\nEnter Initial amount
to be deposited(Min. limit is Rs.1000): ");
            money = input.nextDouble();
            if(money<1000)
                System.out.print("\nCannot open
account for balance less than Rs.1000");
            else
                flag=1;
        }
        CurrentAccount account1 = new
CurrentAccount(name, "Current Account", money);

        while(choice!=4)
        {
            System.out.print("\n\n ----Account Menu--
--");

            System.out.print("\n\n1.Deposit money");
            System.out.print("\n2.Withdraw money");
            System.out.print("\n3.Get mini statement");
            System.out.print("\n4.exit");
            System.out.print("\n\nChoice: ");
            choice = input.nextInt();

            switch(choice)
            {
                case 1: System.out.print("\n\nEnter Amount to be
deposited: ");
                    money = input.nextDouble();
                    System.out.print("\nEnter time since last
transaction(in months): ");
                    time = input.nextInt();
                    account1.deposit(money,time);
                    break;

                case 2: System.out.print("\n\nEnter Amount to be
withdrawn: ");
                    money = input.nextDouble();
                    System.out.print("\nEnter time since last
transaction(in months): ");
                    time = input.nextInt();
                    account1.withdraw(money,time);
                    break;

                case 3: account1.showDetails();
                    break;

                case 4: break;

                default: System.out.print("\n\n Wrong Choice!
\nTerminating...\n");
                    System.exit(0);
            }
        }
    }
}

```

```

        }
    }

    break;

    case 2: System.out.print("\n\nEnter Customer Name:
");
        name = input.nextLine();
        System.out.print("\nEnter Initial amount to
be deposited(Min. limit is Rs.1000): ");
        money = input.nextDouble();
        SavingsAccount account2 = new
SavingsAccount(name,"Saving Account",money);

        while(choice!=4)
        {
            System.out.print("\n\n ----Account Menu--
--");
            System.out.print("\n\n1.Deposit money");
            System.out.print("\n2.Withdraw money");
            System.out.print("\n3.Get mini statement");
            System.out.print("\n4.exit");
            System.out.print("\n\nChoice: ");
            choice = input.nextInt();

            switch(choice)
            {
                case 1: System.out.print("\n\nEnter Amount to be
deposited: ");
                    money = input.nextDouble();
                    System.out.print("\nEnter time since last
transaction(in months): ");
                    time = input.nextInt();
                    account2.deposit(money,time);
                    break;

                case 2: System.out.print("\n\nEnter Amount to be
withdrawn: ");
                    money = input.nextDouble();
                    System.out.print("\nEnter time since last
transaction(in months): ");
                    time = input.nextInt();
                    account2.withdraw(money,time);
                    break;

                case 3: account2.showDetails();
                    break;

                case 4: break;

                default: System.out.print("\n\n Wrong Choice!
\nTerminating...\n");
                    System.exit(0);
            }
        }
    }
}

```

```

                break;

                default: System.out.print("\n\n Wrong Choice!
\nTerminating...\n");
                System.exit(0);

            }
        }
    }
}

```

```

//CO24557 Assignment 3 - problem 5
//Employee management system - shows leave balance and salary since
//DOJ

```

```

import java.util.*;
import java.time.LocalDate;
import Date.*;          //user-defined package

class Employee
{
    double leavesPerMonth;
    double initialSalary;
    int hike;

    public Employee() {}

    public void showDetails(MyDate doj,float leaves)
    {
        LocalDate today = LocalDate.now();          //for
today's date
        int cur_year = today.getYear();
        int cur_month = today.getMonthValue();

        int yearsOfWork = cur_year - doj.getYear();
        if(cur_month-doj.getMonth() < 0)
            yearsOfWork--;

        while(yearsOfWork>0)
        {
            initialSalary = initialSalary +
(initialSalary*hike/100);
            yearsOfWork--;
        }

        System.out.printf("\n\nSalary of employee per month now:
Rs. %.3f lakhs",initialSalary/12);
        System.out.print("\nLeaves remaining =
"+((leavesPerMonth*12)-leaves));
    }
}

```

```

class Manager extends Employee
{
    public Manager()
    {
        initialSalary = 12.0;        //in lpa
        leavesPerMonth = 2.5;
    }
}

class Engineer extends Employee
{
    public Engineer() {}
}

class Researcher extends Employee
{
    public Researcher()
    {
        initialSalary = 12.0;        //in lpa
    }
}

class HR extends Manager
{
    public HR()
    {
        hike = 20;        //percentage
    }
}

class Technology extends Manager
{
    public Technology()
    {
        hike = 30;        //percentage
    }
}

class Developer extends Engineer
{
    public Developer()
    {
        initialSalary = 10.0;        //in lpa
        leavesPerMonth = 3;
        hike = 40;        //percentage
    }
}

class Tester extends Engineer
{
    public Tester()
    {
        initialSalary = 6.0;        //in lpa
    }
}

```

```

        leavesPerMonth = 2.5;
        hike = 10;                                //percentage
    }
}

class Architect extends Engineer
{
    public Architect()
    {
        initialSalary = 12.0;                    //in lpa
        leavesPerMonth = 4;
        hike = 50;                                //percentage
    }
}

class DataScience extends Researcher
{
    public DataScience()
    {
        leavesPerMonth = 4;
        hike = 50;                                //percentage
    }
}

class Analytics extends Researcher
{
    public Analytics()
    {
        leavesPerMonth = 3;
        hike = 40;                                //percentage
    }
}

class Cloud extends Researcher
{
    public Cloud()
    {
        leavesPerMonth = 3;
        hike = 40;                                //percentage
    }
}

public class EmployeeManagementSystem
{
    public static void main(String[] args)
    {
        float leaves;
        MyDate doj = new MyDate();
        Scanner input = new Scanner(System.in);
        int ch;

        System.out.print("\n\nEnter the employee's date of
Joining: ");
        doj.getDate();
    }
}

```

```

        System.out.print("\n\nEnter the Leaves taken for the
current year: ");
        leaves = input.nextFloat();

        System.out.print("\nSelect the designation of Employee:
");

        System.out.print("\n\n1.HR Manager");
        System.out.print("\n2.Technology Manager");
        System.out.print("\n3.Developer");
        System.out.print("\n4.Tester");
        System.out.print("\n5.Architect");
        System.out.print("\n6.Data Science Researcher");
        System.out.print("\n7.Analytics Researcher");
        System.out.print("\n8.Cloud Researcher");
        System.out.print("\n\nChoice: ");

        ch = input.nextInt();

        switch(ch)
        {
            case 1: HR emp1 = new HR();
                    emp1.showDetails(doj,leaves);
                    break;
            case 2: Technology emp2 = new Technology();
                    emp2.showDetails(doj,leaves);
                    break;
            case 3: Developer emp3 = new Developer();
                    emp3.showDetails(doj,leaves);
                    break;
            case 4: Tester emp4 = new Tester();
                    emp4.showDetails(doj,leaves);
                    break;
            case 5: Architect emp5 = new Architect();
                    emp5.showDetails(doj,leaves);
                    break;
            case 6: DataScience emp6 = new DataScience();
                    emp6.showDetails(doj,leaves);
                    break;
            case 7: Analytics emp7 = new Analytics();
                    emp7.showDetails(doj,leaves);
                    break;
            case 8: Cloud emp8 = new Cloud();
                    emp8.showDetails(doj,leaves);
                    break;
            default: System.out.print("\nWrong Choice!
\nTerminating...\n");
                    System.exit(0);
        }
    }
}

```

-----*****-----