Name:S.Geethan Amarthiya

Reg no: 9919004086

Date :15/9/20

Program with coding:

1.Write a java program to implement inheritance using super keyword.

class Box{

private double width;

private double height;

private double length;

public Box(double w, double h, double l)

{

width =w; height=h;length=l;

}

public Box(){

width=-1;height=-1;length=-1;

}

double volume(){

return width\* height\*length;

}

}

class BoxWeight extends Box

{

double weight;

BoxWeight(double w, double h, double l,double we){

super(w,h,l);// call super class parameterized constructor

weight=we;

}

BoxWeight(){

super();//call base class default

}

}

public class main{

public static void main(String[] args){

BoxWeight b1=new BoxWeight(2.9,4.5,2.6,7.8);

BoxWeight b2=new BoxWeight();

double d=b1.volume();

System.out.println("Volume of b1= "+d);

d=b2.volume();

System.out.println("Volume of b2= "+d);

}

}

**2.** Create a class called Date that includes three pieces of information as instance

variables—a month (typeint), a day (typeint) and a year (typeint). Your class

should have a constructor that initializes the three instance variables and

assumes that the values provided are correct. Provide a set and a get method for

each instance variable. Provide a method displayDate that displays the month,

day and year separated by forward slashes(/). Write a test application named

DateTest that demonstrates cl

class Date {

int day ;

int month ;

int year ;

public Date ( int d , int m , int y) {

if(m<13 && d<31){

month = m; day=d; year=y;

}

else{

System.out.println("incorrect date");

}

}

void setMonth(int m){

if(m<13)

month=m;

else

System.out.println("incorrect format");

}

void setDay(int d){

if(d<31)

day=d;

else

System.out.println("incorrect format");

}

void setYear(int y){

if((y/10000)==0)

year=y;

else

System.out.println("incorrect format");

}

int getMonth(){

return month;

}

int getDay(){

return day;

}

int getYear(){

return year;

}

void display () {

System.out.println("The date is " + day +"/" + month + "/" + year);

}

}

public class Main

{

public static void main(String[] args) {

Date d1 = new Date(16,9,2020);

d1.display();

d1.setDay(15);

d1.setMonth(9);

d1.setYear(2020);

}

}

**3. Create class SavingsAccount. Usea static variable annualInterestRate to store the**

**annual interest rate for all account holders. Each object of the class contains a**

**private instance variable savingsBalance indicating the amount the saver**

**currently has ondeposit. Provide method calculateMonthlyInterest to calculate**

**the monthly interest by multiplying the savingsBalance by annualInterestRate**

**divided by 12 this interest should be added to savingsBalance. Provide a static**

**method modifyInterestRate th**

**Provide a static**

**method modifyInterestRate that sets the annualInterestRate to a new value.**

**Write a program to test class SavingsAccount. Instantiate two savingsAccount**

**objects, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively.**

**Set annualInterestRate to 4%, then calculate the monthly interest and print the**

**new balances for both savers. Then set the annualInterestRate to 5%, calculate the**

**next month’s interest and print the new balances for both savers.**

class SavingsAccount{

static float AnnualIntrestrate = (float)4.0;

private float SavingsBalance;

void caluclateMonthlyIntrest(){

float intrest = ((SavingsBalance\*AnnualIntrestrate)/12);

SavingsBalance+=intrest;

System.out.println("balance is " + SavingsBalance);

}

static void ModifyIntrestrate(float rate){

AnnualIntrestrate=rate;

}

public SavingsAccount(float balance){

SavingsBalance=balance;

}

}

public class Main

{

public static void main(String[] args) {

SavingsAccount s1 = new SavingsAccount(2000.0f);

SavingsAccount s2 = new SavingsAccount(3000.0f);

s1.caluclateMonthlyIntrest();

s2.caluclateMonthlyIntrest();

SavingsAccount.ModifyIntrestrate(5.0f);

s1.caluclateMonthlyIntrest();

s2.caluclateMonthlyIntrest();

}

}

**4.** Write a java program create a class called book and initialize the respective details of book using class constructor and access them by creating objects and perform required operations.

import java.util.Scanner;

class Book

{

String bookName;

String author;

String ISBN, publisher;

Book(String title, String auth, String isbn, String publish)

{

bookName = title;

author =auth;

this.ISBN = isbn;

publisher = publish;

}

void setTitle(String name)

{ bookName = name; }

void setAuthor(String auth)

{ author = auth; }

void setISBN(String s)

{ ISBN = s; }

void setPublisher(String p)

{

publisher = p;

}

String getTitle()

{ return bookName; }

String getAuthor()

{ return author; }

String getISBN()

{ return ISBN; }

String getPublisher()

{ return publisher; }

String bookInfo()

{

String info = bookName + " " + author + " " + ISBN + " " + publisher;

return info;

}

}

public class Main

{

public static void main(String[] args) {

Book b[] = new Book[30];

b[0] = new Book("Programming in Java", "Rama", "12345", "Wiley");

String title, auth, isbn, publisher;

Scanner s = new Scanner(System.in);

for (int i =1; i < 5; i++)

{

title = s.next();

auth = s.next();

isbn = s.next();

publisher = s.next();

b[i] = new Book(title,auth,isbn,publisher);

}

b[2].setTitle("Software Testing");

System.out.println(b[2].getTitle());

String info;

for (int i =0; i<5; i++) {

info = b[i].bookInfo();

System.out.println(info);

}

}

}