***1.Create an abstract class Area with methods calc\_area and calc\_volume. Derive 4 classes Sphere(radius) , Cone(radius, height) and Cylinder(radius, height), Box(length, breadth, height) from it. Calculate area and volume of all. (Use Method overriding).***

***(using switchcase)***

import java.util.\*;

abstract class area

{

abstract void cal\_area();

abstract void cal\_vol();

}

class circle1 extends area

{

double r,A,V;

circle1(double r)

{

this.r=r;

}

public void cal\_area()

{

A=3.14\*r\*r;

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=(4/3)\*3.14\*r\*r\*r;

System.out.println("volume="+V);

}

}

class cone extends area

{

double r,h,A,V;

cone(double r,double h)

{

this.r=r;

this.h=h;

}

public void cal\_area()

{

A=3.14\*r\*(r+h);

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=(3.14)\*r\*r\*h/3;

System.out.println("volume="+V);

}

}

class cylinder extends area

{

double r,h,A,V;

cylinder(double r,double h)

{

this.r=r;

this.h=h;

}

public void cal\_area()

{

A=2\*3.14\*r\*h + 2\*3.14\*r\*r;

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=3.14\*r\*r\*h;

System.out.println("volume="+V);

}

}

class sphere extends area

{

double r,A,V;

sphere(double r)

{

this.r=r;

}

public void cal\_area()

{

A=4\*3.14\*r\*r\*r;

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=(4/3)\*3.14\*r\*r\*r;

System.out.println("volume="+V);

}

}

class box extends area

{

double l,b,h,A,V;

box(double l,double b,double h)

{

this.l=l;

this.b=b;

this.h=h;

}

public void cal\_area()

{

A=l\*b;

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=l\*b\*h;

System.out.println("volume="+V);

}

}

public class Main {

public static void main(String[] args)

{

Double r,h,l,b;

Scanner sc=new Scanner(System.in);

System.out.println("enter r");

r=sc.nextDouble();

System.out.println("enter h");

h=sc.nextDouble();

System.out.println("enter the l and b and h: ");

l=sc.nextDouble();

b=sc.nextDouble();

h=sc.nextDouble();

do{

System.out.println("1.circle/t2.cone/t3.cylinder/t4.sphere/t5.box/t6.exit");

System.out.println("enter your choice:");

int ch=sc.nextInt();

switch(ch)

{

case 1:

circle1 c1=new circle1(r);

c1.cal\_area();

c1.cal\_vol();

break;

case 2:

cone c2=new cone(r,h);

c2.cal\_area();

c2.cal\_vol();

break;

case 3:

cylinder c3=new cylinder(r,h);

c3.cal\_area();

c3.cal\_vol();

break;

case 4:

sphere c4=new sphere(r);

c4.cal\_area();

c4.cal\_vol();

break;

case 5:

box c5=new box(l,b,h);

c5.cal\_area();

c5.cal\_vol();

break;

case 6:

System.exit(1);

break;

default:

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

break;

}

}while(true);

}

}

Output:

enter r

2

enter h

3

enter the l and b and h:

4

5

6

1.circle/t2.cone/t3.cylinder/t4.sphere/t5.box/t6.exit

enter your choice:

1

Arae=12.56

volume=25.12

1.circle/t2.cone/t3.cylinder/t4.sphere/t5.box/t6.exit

enter your choice:

2

Arae=50.24

volume=25.12

1.circle/t2.cone/t3.cylinder/t4.sphere/t5.box/t6.exit

enter your choice:

3

Arae=100.48

volume=75.36

1.circle/t2.cone/t3.cylinder/t4.sphere/t5.box/t6.exit

enter your choice:

4

Arae=100.48

volume=25.12

1.circle/t2.cone/t3.cylinder/t4.sphere/t5.box/t6.exit

enter your choice:

5

Arae=20.0

volume=120.0

1.circle/t2.cone/t3.cylinder/t4.sphere/t5.box/t6.exit

enter your choice:

6

***(without switchcase)***

import java.util.\*;

abstract class area

{

abstract void cal\_area();

abstract void cal\_vol();

}

class circle1 extends area

{

double r,A,V;

circle1(double r)

{

this.r=r;

}

public void cal\_area()

{

A=3.14\*r\*r;

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=(4/3)\*3.14\*r\*r\*r;

System.out.println("volume="+V);

}

}

class cone extends area

{

double r,h,A,V;

cone(double r,double h)

{

this.r=r;

this.h=h;

}

public void cal\_area()

{

A=3.14\*r\*(r+h);

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=(3.14)\*r\*r\*h/3;

System.out.println("volume="+V);

}

}

class cylinder extends area

{

double r,h,A,V;

cylinder(double r,double h)

{

this.r=r;

this.h=h;

}

public void cal\_area()

{

A=2\*3.14\*r\*h + 2\*3.14\*r\*r;

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=3.14\*r\*r\*h;

System.out.println("volume="+V);

}

}

class sphere extends area

{

double r,A,V;

sphere(double r)

{

this.r=r;

}

public void cal\_area()

{

A=4\*3.14\*r\*r\*r;

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=(4/3)\*3.14\*r\*r\*r;

System.out.println("volume="+V);

}

}

class box extends area

{

double l,b,h,A,V;

box(double l,double b,double h)

{

this.l=l;

this.b=b;

this.h=h;

}

public void cal\_area()

{

A=l\*b;

System.out.println("Arae="+A);

}

public void cal\_vol()

{

V=l\*b\*h;

System.out.println("volume="+V);

}

}

public class Main {

public static void main(String[] args)

{

Double r,h,l,b;

//area a1=new area();

Scanner sc=new Scanner(System.in);

System.out.println("enter r");

r=sc.nextDouble();

circle1 c1=new circle1(r);

c1.cal\_area();

c1.cal\_vol();

System.out.println("enter h");

h=sc.nextDouble();

cone c2=new cone(r,h);

c2.cal\_area();

c2.cal\_vol();

cylinder c3=new cylinder(r,h);

c3.cal\_area();

c3.cal\_vol();

sphere c4=new sphere(r);

c4.cal\_area();

c4.cal\_vol();

System.out.println("enter the l and b: ");

l=sc.nextDouble();

b=sc.nextDouble();

box c5=new box(l,b,h);

c5.cal\_area();

c5.cal\_vol();

}

}

***Output:***

enter r

2

Arae=12.56

volume=25.12

enter h

3

Arae=31.400000000000002

volume=12.56

Arae=62.8

volume=37.68

Arae=100.48

volume=25.12

enter the l and b:

3

4

Arae=12.0

volume=36.0

**2. Define an abstract class Staff with protected members id and name. Define a parameterized constructor. Define one subclass OfficeStaff with member department. Create n objects of OfficeStaff and display all details.**

import java.util.\*;

abstract class Staff {

protected int id;

protected String name;

public Staff(int id, String name) {

this.id = id;

this.name = name;

}

public abstract void displayDetails();

}

class OfficeStaff extends Staff {

String department;

public OfficeStaff(int id, String name, String department) {

super(id, name);

this.department = department;

}

@Override

public void displayDetails() {

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

System.out.println("Name: " + name);

System.out.println("Department: " + department);

}

}

public class Main {

public static void main(String[] args) {

int n ,i,id;

String name,department;

Scanner sc=new Scanner(System.in);

System.out.println("enter the number of records:");

n=sc.nextInt();

OfficeStaff[] office1= new OfficeStaff[n];

for ( i = 0; i < n; i++) {

System.out.println("OfficeStaff " + (i + 1) + " details:");

System.out.println("enter id name and department:");

id=sc.nextInt();

name=sc.next();

department=sc.next();

office1[i]=new OfficeStaff(id,name,department);

}

System.out.println("\n displayDetails");

for(i=0;i<n;i++)

{

System.out.println("OfficeStaff " + (i + 1) + " details:");

office1[i].displayDetails();

System.out.println();

}

}

}

***Output:***

enter the number of records:

2

OfficeStaff 1 details:

enter id name and department:

101

nita

bcs

OfficeStaff 2 details:

enter id name and department:

2

minal

bca

displayDetails

OfficeStaff 1 details:

ID: 101

Name: nita

Department: bcs

OfficeStaff 2 details:

ID: 2

Name: minal

Department: bca

***3. Create an abstract class “order” having members id,description and add abstract method show().Create two subclasses “Purchase Order” and “Sales Order” having members Vendor name and customer name respectively.Define methods accept and display in all cases. Create 3 objects each of Purchase Order and Sales Order and accept and display details***

import java.util.Scanner;

abstract class Order {

protected int id;

protected String description;

public abstract void show();

public abstract void accept(Scanner scanner);

}

class PurchaseOrder extends Order {

private String vendorName;

@Override

public void accept(Scanner scanner) {

System.out.print("Enter Purchase Order ID: ");

id = scanner.nextInt();

System.out.print("Enter Purchase Order Description: ");

description = scanner.nextLine();

System.out.print("Enter Vendor Name: ");

vendorName = scanner.nextLine();

}

@Override

public void show() {

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

System.out.println("Description: " + description);

System.out.println("Vendor Name: " + vendorName);

}

}

class SalesOrder extends Order {

private String customerName;

@Override

public void accept(Scanner scanner) {

System.out.print("Enter Sales Order ID: ");

id = scanner.nextInt();

System.out.print("Enter Sales Order Description: ");

description = scanner.nextLine();

System.out.print("Enter Customer Name: ");

customerName = scanner.nextLine();

}

@Override

public void show() {

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

System.out.println("Description: " + description);

System.out.println("Customer Name: " + customerName);

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

PurchaseOrder[] purchaseOrders = new PurchaseOrder[3];

for (int i = 0; i < 3; i++) {

purchaseOrders[i] = new PurchaseOrder();

System.out.println("Enter details for Purchase Order " + (i + 1) + ":");

purchaseOrders[i].accept(scanner);

}

SalesOrder[] salesOrders = new SalesOrder[3];

for (int i = 0; i < 3; i++) {

salesOrders[i] = new SalesOrder();

System.out.println("Enter details for Sales Order " + (i + 1) + ":");

salesOrders[i].accept(scanner);

}

System.out.println("\nDisplaying Purchase Orders:");

for (int i = 0; i < 3; i++) {

System.out.println("Purchase Order " + (i + 1) + " details:");

purchaseOrders[i].show();

System.out.println();

}

System.out.println("Displaying Sales Orders:");

for (int i = 0; i < 3; i++) {

System.out.println("Sales Order " + (i + 1) + " details:");

salesOrders[i].show();

System.out.println();

}

}

}

***Output:***

Enter details for Purchase Order 1:

Enter Purchase Order ID: 1

Enter Purchase Order Description: DRESS

Enter Vendor Name: VIDYA

Enter details for Purchase Order 2:

Enter Purchase Order ID: 2

Enter Purchase Order Description: MINAL

Enter Vendor Name: CREAM

Enter details for Purchase Order 3:

Enter Purchase Order ID: 3

Enter Purchase Order Description: LIPSTIC

Enter Vendor Name: MITALI

Enter details for Sales Order 1:

Enter Sales Order ID: 4

Enter Sales Order Description: LEHENGA

Enter Customer Name: VISHU

Enter details for Sales Order 2:

Enter Sales Order ID: 5

Enter Sales Order Description: SAREE

Enter Customer Name: CHIU

Enter details for Sales Order 3:

Enter Sales Order ID: 6

Enter Sales Order Description: EYELINER

Enter Customer Name: SHIVANI

Displaying Purchase Orders:

Purchase Order 1 details:

Purchase Order ID: 1

Description: DRESS

Vendor Name: VIDYA

Purchase Order 2 details:

Purchase Order ID: 2

Description: MINAL

Vendor Name: CREAM

Purchase Order 3 details:

Purchase Order ID: 3

Description: LIPSTIC

Vendor Name: MITALI

Displaying Sales Orders:

Sales Order 1 details:

Sales Order ID: 4

Description: LEHENGA

Customer Name: VISHU

Sales Order 2 details:

Sales Order ID: 5

Description: SAREE

Customer Name: CHIU

Sales Order 3 details:

Sales Order ID: 6

Description: EYELINER

Customer Name: SHIVANI

***4. A bank Maintains two kinds of accounts - Savings Account and Current Account. The savings account provides compound interest, deposit and withdrawal facilities. The current account only provides deposit and withdrawal facilities. Current account holders should also Maintain a minimum balance. If balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number, and type of account. From this derive the classes Curr-acct and Sav-acct. Include the necessary methods in order to achieve the following tasks. a. Accept deposit from a customer and update the balance. b. Display the balance. c. Compute interest and add to balance. d. Permit withdrawal and update the balance ( Check for the minimum balance, impose penalty if necessary).***

import java.util.Scanner;

class Account {

protected String customerName;

protected long accountNumber;

protected String accountType;

protected double balance;

public Account(String customerName, long accountNumber, String accountType, double initialBalance) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = initialBalance;

}

public void deposit(double amount) {

balance += amount;

System.out.println("Deposit of $" + amount + " successful.");

}

public void displayBalance() {

System.out.println("Account Balance: $" + balance);

}

public void computeInterest() {

}

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawal of $" + amount + " successful.");

} else {

System.out.println("Insufficient balance for withdrawal.");

}

}

}

class CurrentAccount extends Account {

public CurrentAccount(String customerName, long accountNumber, double initialBalance) {

super(customerName, accountNumber, "Current Account", initialBalance);

}

@Override

public void computeInterest() {

System.out.println("No interest is earned on Current Account.");

}

}

class SavingsBankAccount extends Account {

private double interestRate;

private double minimumBalance;

public SavingsBankAccount(String customerName, long accountNumber, double initialBalance,

double interestRate, double minimumBalance) {

super(customerName, accountNumber, "Savings Bank Account", initialBalance);

this.interestRate = interestRate;

this.minimumBalance = minimumBalance;

}

@Override

public void computeInterest() {

double interest = balance \* (interestRate / 100);

balance += interest;

System.out.println("Interest of $" + interest + " earned.");

}

@Override

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawal of $" + amount + " successful.");

} else {

System.out.println("Insufficient balance for withdrawal.");

}

if (balance < minimumBalance) {

double penalty = 0.1 \* minimumBalance;

balance -= penalty;

System.out.println("Service tax of $" + penalty + " imposed due to balance below minimum.");

}

}

}

class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter customer name:");

String customerName = scanner.nextLine();

System.out.println("Enter account number:");

long accountNumber = scanner.nextLong();

System.out.println("Enter initial balance:");

double initialBalance = scanner.nextDouble();

CurrentAccount currentAccount = new CurrentAccount(customerName, accountNumber, initialBalance);

System.out.println("Enter deposit amount:");

double depositAmount = scanner.nextDouble();

currentAccount.deposit(depositAmount);

currentAccount.displayBalance();

currentAccount.computeInterest();

System.out.println("Enter withdrawal amount:");

double withdrawalAmount = scanner.nextDouble();

currentAccount.withdraw(withdrawalAmount);

currentAccount.displayBalance();

System.out.println("Enter savings account interest rate:");

double interestRate = scanner.nextDouble();

System.out.println("Enter minimum balance for savings account:");

double minimumBalance = scanner.nextDouble();

SavingsBankAccount savingsAccount = new SavingsBankAccount(customerName, accountNumber, initialBalance,

interestRate, minimumBalance);

System.out.println("Enter deposit amount for savings account:");

depositAmount = scanner.nextDouble();

savingsAccount.deposit(depositAmount);

savingsAccount.displayBalance();

savingsAccount.computeInterest();

System.out.println("Enter withdrawal amount for savings account:");

withdrawalAmount = scanner.nextDouble();

savingsAccount.withdraw(withdrawalAmount);

savingsAccount.displayBalance();

}

}

***Output:***

Enter customer name:

mira

Enter account number:

23

Enter initial balance:

10000

Enter deposit amount:

2000

Deposit of $2000.0 successful.

Account Balance: $12000.0

No interest is earned on Current Account.

Enter withdrawal amount:

2300

Withdrawal of $2300.0 successful.

Account Balance: $9700.0

Enter savings account interest rate:

10

Enter minimum balance for savings account:

10000

Enter deposit amount for savings account:

2000

Deposit of $2000.0 successful.

Account Balance: $12000.0

Interest of $1200.0 earned.

Enter withdrawal amount for savings account:

2300

Withdrawal of $2300.0 successful.

Account Balance: $10900.0