***1.area***

import java.util.\*;

class Area

{

Double r,A;

Area(double r)

{

this.r=r;

}

void cal\_area()

{

A=3.14\*r\*r;

System.out.println("area:"+A);

}

}

class Volume extends Area{

double h,v;

Volume(double r, double h )

{

super(r);

this.h=h;

}

void cal\_vol(){

cal\_area();

v=A\*h;

System.out.println("Volume:"+v);

}

}

class volume1 extends Volume

{

double l,v1;

volume1(double r,double h,double l)

{

super(r,h);

this.l=l;

}

void cal\_vol1()

{

cal\_vol();

v1=v\*l;

System.out.println("Volume1="+v1);

}

}

public class Main

{

public static void main(String[] args)

{

double r,h,l;

Scanner sc=new Scanner (System.in);

System.out.println("Enter r & h l");

r=sc.nextDouble();

h=sc.nextDouble();

l=sc.nextDouble();

volume1 v1=new volume1(r,h,l);

v1.cal\_vol1();

}

}

***Output:***

Enter r & h l

2

3

4

area:12.56

Volume:37.68

Volume1=150.72

***2.student multilevel inheritance.***

import java.util.\*;

class student

{

int id;

String name;

student(int id, String name)

{

this.id=id;

this.name=name;

}

void display()

{

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

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

}

}

class marks extends student

{

double m1,m2,m3;

marks(int id,String name,double m1,double m2, double m3)

{

super(id,name);

this.m1=m1;

this.m2=m2;

this.m3=m3;

}

void show()

{

super.display();

System.out.println("m1:"+m1);

System.out.println("m2:"+m2);

System.out.println("m3:"+m3);

}

}

class result extends marks

{

double per;

double total;

result(int id,String name,double m1,double m2,double m3)

{

super(id,name,m1,m2,m3);

}

void cal\_result()

{

total=m1+m2+m3;

per=(total/300)\*100;

System.out.println("total:"+total);

System.out.println("per:"+per);

}

void cal\_gread()

{

if(per>=90)

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

else if(per>=80)

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

else if(per>=70)

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

else if(per>=60)

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

else if(per>=40)

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

else

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

}

}

public class Main

{

public static void main(String[] args)

{

double m1,m2,m3;

int id;

String name;

Scanner sc=new Scanner (System.in);

System.out.println("Enter id name m1,m2,m3");

id=sc.nextInt();

name=sc.next();

m1=sc.nextDouble();

m2=sc.nextDouble();

m3=sc.nextDouble();

result v1=new result(id,name,m1,m2,m3);

v1.cal\_result();

v1.cal\_gread();

}

}

***Output:***

Enter id name m1,m2,m3

1

ritu

67

98

78

total:243.0

per:81.0

Grade B

***3.hirarchical***

***Employee full time(n of hour,hr\_rate,salary) and part time(n of day,day\_rate,salary).***

import java.util.Scanner;

class Emp12

{

double id;

String name,desg;

Emp12(Double id,String name,String desg)

{

this.id=id;

this.name=name;

this.desg=desg;

}

public void display()

{

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

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

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

}

}

class part\_time extends Emp12

{

double hr\_rate,sal;

int n\_hr;

part\_time(Double id,String name,String desg,int n\_hr,double hr\_rate)

{

super( id,name,desg);

this.n\_hr=n\_hr;

this.hr\_rate=hr\_rate;

}

public void cal\_sal()

{

System.out.println("No of hr:"+n\_hr);

System.out.println("Hr rate:"+hr\_rate);

sal=(n\_hr\*hr\_rate);

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

}

}

class Full\_time extends Emp12

{

double day\_rate,sal;

int n\_day;

Full\_time(Double id,String name,String desg,int n\_day,double day\_rate)

{

super(id,name,desg);

this.n\_day=n\_day;

this.day\_rate=day\_rate;

}

public void cal\_sal()

{

System.out.println("No of hr:"+n\_day);

System.out.println("Hr rate:"+day\_rate);

System.out.println("Salary:"+(n\_day\*day\_rate));

}

}

public class Main

{

public static void main(String[] args)

{

double id;

String name,desg;

double hr\_rate;

int n\_hr,ch;

double day\_rate;

int n\_day;

Scanner sc=new Scanner(System.in);

do

{

System.out.println("1:Accept details of part time empl");

System.out.println("2:Accept details of full time empl");

System.out.println("3:Exit");

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

ch = sc.nextInt();

switch(ch)

{

case 1:

System.out.println("Enter id,name & desg,n0f hr,hr rate");

id=sc.nextDouble();

name=sc.next();

desg=sc.next();

n\_hr=sc.nextInt();

hr\_rate=sc.nextDouble();

part\_time p1=new part\_time(id, name, desg, n\_hr, hr\_rate);

p1.display();

p1.cal\_sal();

break;

case 2:

System.out.println("Enter id,name & desg,n0f day,day rate");

id=sc.nextDouble();

name=sc.next();

desg=sc.next();

n\_day=sc.nextInt();

day\_rate=sc.nextDouble();

Full\_time f1=new Full\_time(id, name, desg, n\_day, day\_rate);

f1.display();

f1.cal\_sal();

break;

case 3:

System.exit(0);

break;

default:System.out.println("Invalid choice");

}

}while(ch<=3);

}

}

***Output:***

1:Accept details of part time empl

2:Accept details of full time empl

3:Exit

enter u r choice

1

Enter id,name & desg,n0f hr,hr rate

1

ritesh

tester

5

1000

id:1.0

name :ritesh

Designation :tester

No of hr:5

Hr rate:1000.0

Salary:5000.0

1:Accept details of part time empl

2:Accept details of full time empl

3:Exit

enter u r choice

2

Enter id,name & desg,n0f day,day rate

2

minal

developer

23

25000

id:2.0

name :minal

Designation :developer

No of hr:23

Hr rate:25000.0

Salary:575000.0

1:Accept details of part time empl

2:Accept details of full time empl

3:Exit

enter u r choice

3

***4. Write a Java program to create a super class Vehicle having members Company and price. Derive 2 different classes LightMotorVehicle (members – mileage) and HeavyMotorVehicle (members – capacity-in-tons). Accept the information for n vehicles and display the information in appropriate form. While taking data, ask the user about the type of vehicle first.(n no of object)***

import java.util.Scanner;

class Vehicle {

String company;

double price;

Vehicle(String company, double price) {

this.company = company;

this.price = price;

}

void display() {

System.out.println("Company: " + company);

System.out.println("Price: " + price);

}

}

class LightMotorVehicle extends Vehicle {

double mileage;

LightMotorVehicle(String company, double price, double mileage) {

super(company, price);

this.mileage = mileage;

}

void display() {

super.display();

System.out.println("Mileage: " + mileage + " km/l");

System.out.println("Company: " + company);

System.out.println("Price: " + price);

}

}

class HeavyMotorVehicle extends Vehicle {

double capacityInTons;

HeavyMotorVehicle(String company, double price, double capacityInTons) {

super(company, price);

this.capacityInTons = capacityInTons;

}

void display() {

super.display();

System.out.println("Capacity: " + capacityInTons + " tons");

System.out.println("Company: " + company);

System.out.println("Price: " + price);

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int m,p;

System.out.print("Enter the number of vehicles: ");

int n = scanner.nextInt();

scanner.nextLine(); // consume the newline character

Vehicle[] vehicles = new Vehicle[n];

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

System.out.println("Enter details for vehicle " + (i + 1));

System.out.print("Enter type of vehicle (1 for Light Motor Vehicle, 2 for Heavy Motor Vehicle): ");

int type = scanner.nextInt();

scanner.nextLine(); // consume the newline character

switch (type) {

case 1:

System.out.print("Enter company: ");

String company = scanner.next();

System.out.print("Enter price: ");

Double price = scanner.nextDouble();

System.out.print("Enter mileage: ");

Double mileage = scanner.nextDouble();

LightMotorVehicle v1[]=new LightMotorVehicle(n);

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

{

v1[i]=new LightMotorVehicle(company, price, mileage);

v1[i].display();

}

break;

case 2:

System.out.print("Enter company: ");

company = scanner.next ();

System.out.print("Enter price: ");

price = scanner.nextDouble();

System.out.print("Enter capacity in tons: ");

Double capacityInTons = scanner.nextDouble();

HeavyMotorVehicle v2[]= new HeavyMotorVehicle(n);

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

{

v2[i]=new HeavyMotorVehicle(company, price, capacityInTons);

v2[i].display();

}

break;

default:

System.out.println("Invalid type! Skipping this vehicle.");

break;

}

}

}

}

***5. Write a program which has class Movie(title,amount,no\_of\_ticket) and inherit***

***Following classes TaxedMovie(tax,finalAmount) and TaxFreeMovie(finalAmount).***

***Use calculateTicketAmount() methodin both subclasses.***

***Create objects of TaxedMovie and TaxFreeMovie in main class using super class reference***

***print movies info with final amount for both the objects.***

import java.util.\*;

class Movie {

String title;

double amount;

int no\_of\_ticket;

Movie(String title, double amount, int no\_of\_ticket) {

this.title = title;

this.amount = amount;

this.no\_of\_ticket = no\_of\_ticket;

}

void display() {

System.out.println("Title: " + title);

System.out.println("Amount per Ticket: " + amount);

System.out.println("Number of Tickets: " + no\_of\_ticket);

}

}

class TaxedMovie extends Movie {

double tax;

double finalAmount;

TaxedMovie(String title, double amount, int no\_of\_ticket, double tax) {

super(title, amount, no\_of\_ticket);

this.tax = tax;

}

double calculateTicketAmount() {

return finalAmount = amount \* no\_of\_ticket \* (1 + tax / 100);

}

void display() {

super.display();

System.out.println("Tax: " + tax + "%");

System.out.println("final amount is"+ calculateTicketAmount());

}

}

class TaxFreeMovie extends Movie {

double finalAmount;

TaxFreeMovie(String title, double amount, int no\_of\_ticket) {

super(title, amount, no\_of\_ticket);

}

double calculateTicketAmount() {

return finalAmount = amount \* no\_of\_ticket;

}

void display() {

super.display();

System.out.println("final amount is"+ calculateTicketAmount());

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter details for the movie:");

System.out.print("Enter title: ");

String title = scanner.nextLine();

System.out.print("Enter amount per ticket: ");

double amount = scanner.nextDouble();

System.out.print("Enter number of tickets: ");

int no\_of\_ticket = scanner.nextInt();

System.out.print("Enter type of movie (1 for Taxed Movie, 2 for Tax-Free Movie): ");

int type = scanner.nextInt();

switch (type) {

case 1:

System.out.print("Enter tax percentage: ");

double tax = scanner.nextDouble();

TaxedMovie m1= new TaxedMovie(title, amount, no\_of\_ticket, tax);

m1.display();

break;

case 2:

TaxFreeMovie m2= new TaxFreeMovie(title, amount, no\_of\_ticket);

m2.display();

break;

default:

System.out.println("Invalid type! Exiting.");

System.exit(0);

}

}

}

***Output:***

Enter details for the movie:

Enter title: gold

Enter amount per ticket: 200

Enter number of tickets: 3

Enter type of movie (1 for Taxed Movie, 2 for Tax-Free Movie): 1

Enter tax percentage: 20

Title: gold

Amount per Ticket: 200.0

Number of Tickets: 3

Tax: 20.0%

final amount is720.0

***6. Write a program for multilevel inheritance such that country is inherited from continent. State is inherited from country. Display the place, state, country and continent.***

import java.util.Scanner;

class Continent {

String continentName;

Continent(String continentName) {

this.continentName = continentName;

}

void display() {

System.out.println("Continent: " + continentName);

}

}

class Country extends Continent {

String countryName;

Country(String continentName, String countryName) {

super(continentName);

this.countryName = countryName;

}

void display() {

super.display();

System.out.println("Country: " + countryName);

}

}

class State extends Country {

String stateName;

State(String continentName, String countryName, String stateName) {

super(continentName, countryName);

this.stateName = stateName;

}

void display() {

super.display();

System.out.println("State: " + stateName);

}

}

class city extends State {

String cityName;

city(String continentName, String countryName, String stateName, String cityName) {

super(continentName, countryName,stateName);

this.cityName = cityName;

}

void display() {

super.display();

System.out.println("Place: " + cityName);

}

}

class place extends city {

String placeName;

place(String continentName, String countryName, String stateName, String cityName ,String placeName) {

super(continentName, countryName,stateName,cityName);

this.placeName = placeName;

}

void display() {

super.display();

System.out.println("Place: " + placeName);

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter continent name: ");

String continentName = scanner.nextLine();

System.out.print("Enter country name: ");

String countryName = scanner.nextLine();

System.out.print("Enter state name: ");

String stateName = scanner.nextLine();

System.out.print("Enter city name: ");

String cityName = scanner.nextLine();

System.out.print("Enter place name: ");

String placeName = scanner.nextLine();

place place1 = new place(continentName, countryName, stateName,cityName, placeName);

System.out.println("\nDisplaying Information:");

place1.display();

}

}

***Output:***

Enter continent name: ashiya

Enter country name: india

Enter state name: maharashtra

Enter city name: ahmednagar

Enter place name: karjat

Displaying Information:

Continent: ashiya

Country: india

State: maharashtra

Place: ahmednagar

Place: karjat

***7. Create an abstract class “order” having members id,description.Create two subclasses “Purchase Order” and “Sales Order” having members vendor name and customerr 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 {

int id;

String description;

Order(int id, String description) {

this.id = id;

this.description = description;

}

}

class PurchaseOrder extends Order {

String vendorName;

PurchaseOrder(int id, String description,String vendorName) {

super(id, description);

this.vendorName=vendorName;

}

void displayDetails() {

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

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

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

}

}

class SalesOrder extends Order {

String customerName;

SalesOrder(int id, String description, String customerName) {

super(id, description);

this.customerName=customerName;

}

void displayDetails() {

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);

Order[] orders = new Order[6]; // Array to store 3 PurchaseOrders and 3 SalesOrders

for (int i = 0; i < 6; i++) {

System.out.println("Enter details for order " + (i + 1));

System.out.print("Enter order type (1 for Purchase Order, 2 for Sales Order): ");

int type = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter order ID: ");

int id = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter description: ");

String description = scanner.nextLine();

switch (type) {

case 1:

System.out.print("Enter vendorName: ");

String vendorName = scanner.nextLine();

PurchaseOrder p1 = new PurchaseOrder(id, description,vendorName);

p1.displayDetails();

break;

case 2:

System.out.print("Enter customerName: ");

String customerName = scanner.nextLine();

SalesOrder p2 = new SalesOrder(id, description,customerName);

p2.displayDetails();

break;

default:

System.out.println("Invalid order type! Skipping this order.");

}

}

}

}

***Output:***

Enter details for order 1

Enter order type (1 for Purchase Order, 2 for Sales Order): 1

Enter order ID: 1

Enter description: dress

Enter vendorName: nitya

Purchase Order ID: 1

Description: dress

Vendor Name: nitya

Enter details for order 2

Enter order type (1 for Purchase Order, 2 for Sales Order): 2

Enter order ID: 2

Enter description: mastani

Enter customerName: veera

Sales Order ID: 2

Description: mastani

Customer Name: veera

Enter details for order 3

8. ***Define an 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.Scanner;

class Staff {

protected int id;

protected String name;

Staff(int id, String name) {

this.id = id;

this.name = name;

}

void display() {

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

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

}

}

class OfficeStaff extends Staff {

String department;

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

super(id, name);

this.department = department;

}

void display() {

super.display();

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

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of OfficeStaff objects to create: ");

int n = scanner.nextInt();

OfficeStaff[] staffArray = new OfficeStaff[n];

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

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

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

int id = scanner.nextInt();

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

String name = scanner.next ();

System.out.print("Enter Department: ");

String department = scanner.next ();

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

}

System.out.println("\nDisplaying OfficeStaff Details:");

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

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

staffArray[i].display();

System.out.println();

}

}

}

***Output:***

Enter the number of OfficeStaff objects to create: 2

Enter details for OfficeStaff 1

Enter ID: 1

Enter Name: praniti

Enter Department: bcs

Enter details for OfficeStaff 2

Enter ID: 2

Enter Name: vidya

Enter Department: bca

Displaying OfficeStaff Details:

OfficeStaff 1:

ID: 1

Name: praniti

Department: bcs

OfficeStaff 2:

ID: 2

Name: vidya

Department: bca