**SMD Assignment 1**

#include <iostream>

using namespace std;

**//Question No .1 //**

**/\* Explanation Answer:**

the problem in given question was it will show an error of multple inheritance as getranking funtion was present in both batsman and bowler class so it will not know which getranking will show output. java doesn't support multiple inhertance.\*/

// Below is a problem solution that getranking() should be in cricketer class.//

class Cricketer {

private:

String Name;

int Age;

String Nationality;

int Ranking;

public Cricketer(String Name, int Age, String Nationality, int Ranking) {

this.Name = Name;

this.Age = Age;

this.Nationality = Nationality;

this.Ranking = Ranking;

}

public int getRanking() {

return Ranking;

}

public void print() {

System.out.println("Name: " + Name + "Age: " + Age + "Nationality: " + Nationality);

}

}

class Batsman extends Cricketer {

private int score;

private int BatsmanRanking;

public Batsman(String Name, int Age, String Nationality, int score, int BatsmanRanking) {

super(Name, Age, Nationality,Ranking);

this.score = score;

this.BatsmanRanking = BatsmanRanking;

}

public void print() {

super.print();

System.out.println("Score: " + score);

System.out.println("Batsman Ranking: " + BatsmanRanking);

}

}

class Bowler extends Cricketer {

private int wickets;

private int BowlerRanking;

public Bowler(String Name, int Age, String Nationality, int wickets, int BowlerRanking) {

super(Name, Age, Nationality,Ranking);

this.wickets = wickets;

this.BowlerRanking = BowlerRanking;

}

public void print() {

super.print();

System.out.println("Wickets: " + wickets);

System.out.println("Bowler Ranking: " + BowlerRanking);

}

}

class AllRounder extends Batsman,Bowler {

private int ranking;

public AllRounder(String Name, int Age, String Nationality, int score, int BatsmanRanking,

int wickets, int BowlerRanking

int ranking) {

super(Name, Age, Nationality, score, BatsmanRanking, wickets,BowlerRanking);

this.ranking = ranking;

}

public void print() {

super.print();

System.out.println("All-Rounder Ranking: " + ranking);

}

//Override

public int getRanking() {

return ranking;

}

}

class Main {

public static void main(String[] args) {

Batsman batsman = new Batsman("Baber", 26, "Pakistani", 4000, 1);

Bowler bowler = new Bowler("Shaheen", 25, "Pakistani", 100, 2);

AllRounder allRounder = new AllRounder("Shakib", 32, "Bangladesh", 8000, 3, 5);

System.out.println("Batsman Details:");

batsman.print();

System.out.println("Batsman Ranking: " + batsman.getRanking());

System.out.println("\nBowler Details:");

bowler.print();

System.out.println("Bowler Ranking: " + bowler.getRanking());

System.out.println("\nAll-Rounder Details:");

allRounder.print();

System.out.println("All-Rounder Ranking: " + allRounder.getRanking());

}

}

//Question No 2: //

public abstract class person{

protected:

int AadharId;

string Name;

string Address;

string profession;

int phone;

public abstract string Register(int UID);

public abstract string Perform(string duty);

public abstract string Include(string service);

public abstract string like(int item);

}

public class student extends person{

private:

int studentID;

string major;

int year;

int semester;

string college;

int grade;

public:

student(int AadharID, String Name, String Address, int phone, String profession,

int studentID, String major, int year, int semester, String college, int grade) {

super(AadharID, Name, Address, phone, profession);

this.studentID = studentID;

this.major = major;

this.year = year;

this.semester = semester;

this.college = college;

this.grade = grade;

}

string Register(string Course){

return (super.Name +"registered for the course" + Course );

}

string getgrade(string course){

return (super.Name +"grade in course:"+ course + "is" + grade);

}

string attend(string classs){

return (super.Name +" attend the class: "+ classs);

}

string learn(){

return(super.Name +" is learning")

}

}

public class staff extends person{

private:

String Empid;

String EmpRole

String EmpDept;

float EmpSalary;

int EmpExperience;

public:

staff(int AadharID, String Name, String Address, int phone, String profession,

String Empid, String EmpRole, String EmpDept, float EmpSalary, int EmpExperience)

{

super(AadharID, Name, Address, phone, profession);

this.Empid = Empid;

this.EmpRole = EmpRole;

this.EmpDept = EmpDept;

this.EmpSalary = EmpSalary;

this.EmpExperience = EmpExperience;

}

string attend(string duty){

return (super.Name + " attend his duty of: " + duty );

}

string getPromotion(){

return(super.Name + " employe got the promotion");

}

int getSalary(){

return(super.name + "employe got his salary"+ EmpSalary);

}

}

//Concrete PGStudent class inheriting from Student and person

class PGStudent extends student extends person{

private:

int rollNo;

String specialization;

public:

PGStudent(int AadharID, String Name, String Address, int phone, String profession,

int studentID, String major, int year, int semester, String college, int grade,

int rollNo, String specialization)

{

super(AadharID, Name, Address, phone, profession, studentID, major, year, semester, college, grade);

this.rollNo = rollNo;

this.specialization = specialization;

}

String submitThesis() {

return ("PG Student " + super.Name + " with roll number " + rollNo + " submitted his thesis in " + specialization);

}

}

// Concrete PGStudent class inheriting from Student

class Faculty extends staff {

private:

string facultyspecialization;

string section;

int NoOfStudents;

string qualification;

public:

Faculty(int AadharID, String Name, String Address, int phone, String profession,

String Empid, String EmpRole, String EmpDept, float EmpSalary, int EmpExperience,

string facultyspecialization, string section, int NoOfStudents, string qualification;)

{

super(AadharID, Name, Address, phone, profession, Empid, EmpRole,EmpDept, EmpSalary,EmpExperience);

this.section = section;

this.facultyspecialization = facultyspecialization;

this.NoOfStudents = NoOfStudents;

this.qualification = qualification;

}

String Teach(string course) {

return ("Faculty "super.Name + " teach " + course + " to " + section);

}

String Access(student grade) {

return ("student "+ super.Name + " get: " + grade +"grade");

}

}

class Technician extends staff {

private:

string TechType;

public:

Technician(int AadharID, String Name, String Address, int phone, String profession,

String Empid, String EmpRole, String EmpDept, float EmpSalary, int EmpExperience,

string TechType)

{

super(AadharID, Name, Address, phone, profession,Empid, EmpRole, EmpDept,EmpSalary, EmpExperience);

this.TechType = TechType;

}

String Maintain(string lab) {

return (super.Name + " is " + lab + " technician ");

}

String Install(string system) {

return (super.Name + " installed " + system +"in lab");

}

}