

**Adrian Sanchez Rodriguez**

**EC1939656**

Edinburgh College – Sighthill Campus

**H17135 - Software Development Object Oriented Programming**

Assessment (World Skills Competition WSC)

**Contents**

[**Problem Definition** 3](#_Toc102263779)

[**Requirements** 3](#_Toc102263780)

[**Design** 3](#_Toc102263781)

[**Implementation** 4](#_Toc102263782)

[**Testing** 5](#_Toc102263783)

[**Conclusion and recommendations** 7](#_Toc102263784)

[**Appendix** 8](#_Toc102263785)

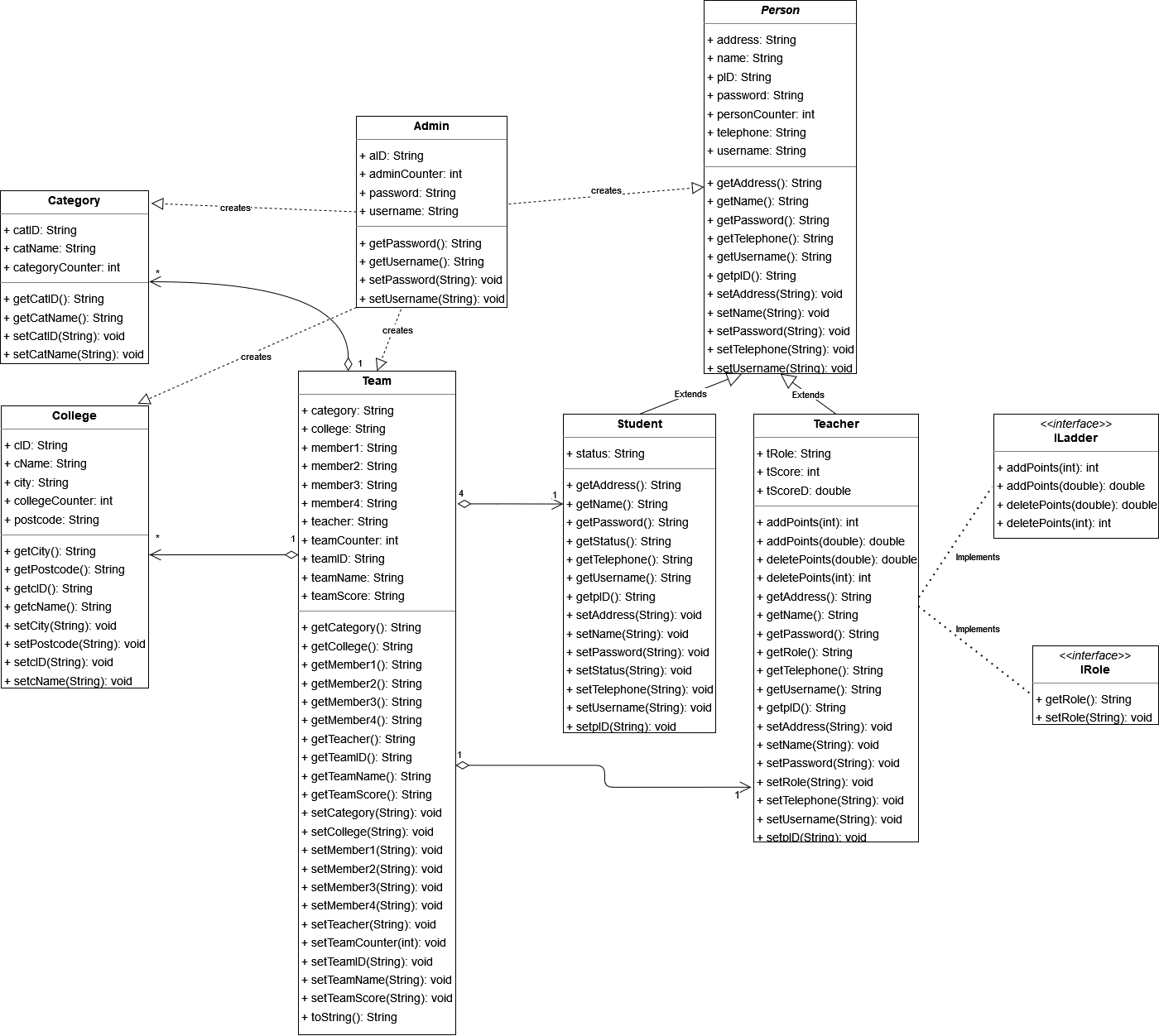
# **Problem Definition**

The client requires a programme that allows recording users (administrators, students, coaches and judges), teams and their scores, to display them in a ranking.

# **Requirements**

* The program will create an administrator account capable of store different pieces of information such as: users, teams, colleges, competition categories and the teams scores for the ranking
* The work frame used for this project is Microsoft Visio for the diagrams, IntelliJ Idea for the IDE and Microsoft Word for the report.
* The program is intended to run on Windows, via command line.
* The testing approach will be White-box, with expected results depending on input.

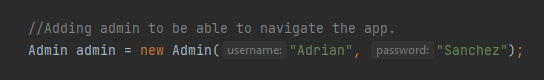
# **Design**



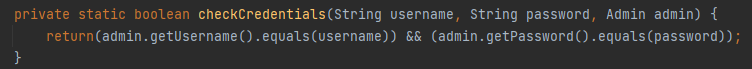
# **Implementation**

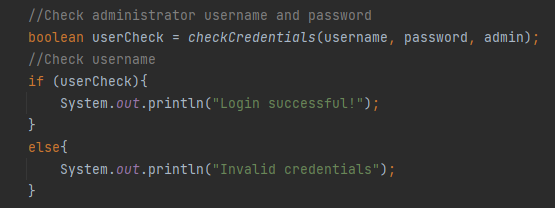
The steps taken were:

1. Generate an administrator that can manage the information in the app



1. Check for user credentials. If correct, allow the login



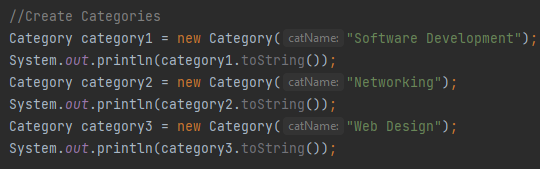


1. Implement objects for colleges, students, teachers, categories, and teams



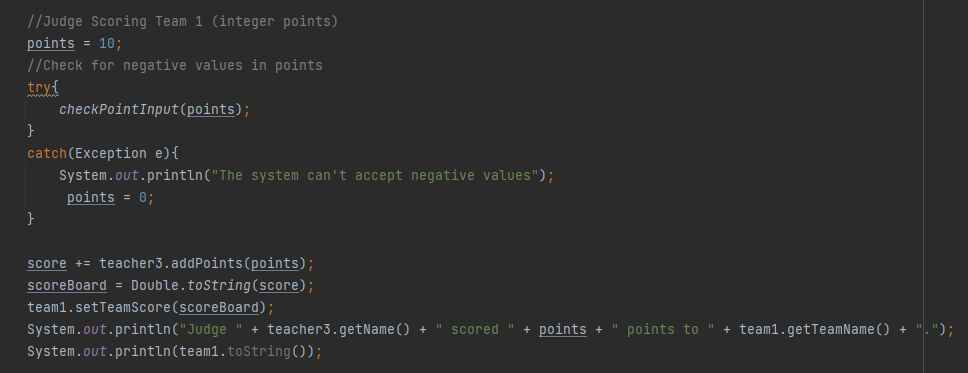




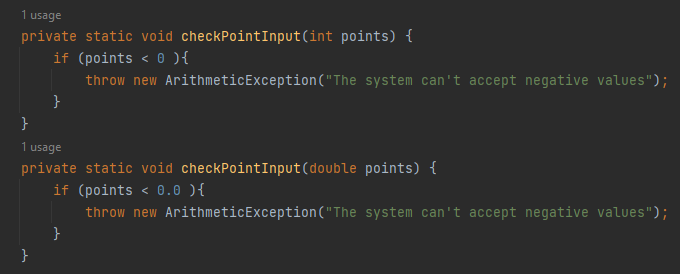




1. When all the information is properly filled up and the competition is taking place, the judges will grant points to the teams in each category. After that, the points will be displayed on the teams score boards.



* Note that an exception handler to prevent negative points to be introduced by the judges is taking place before the point addition.



# **Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test#/**  **Tester/date** | **Entry Value** | **Expect**  **Output** | **Actual Output**  **(Screenshot)** | **Notes** |
| #1 / 28-04-2022 | Admin created and credential checked with right input | Credential verified and displays “login successful!” |  | Program works ok, no problem found. |
| #2 / 28-04-2022 | Admin created and credential checked with wrong input | Credential not verified and displays “invalid credentials” |  | Program works ok, no problem found. |
| #3 / 28-04-2022 | Create students with variable fields properly filled | The object is created and displays correctly on screen |  | Program works ok, no problem found. |
| #4 / 28-04-2022 | Create students with missing variable fields | Compiler error |  | Exception handler needed to check whether the object is being created with all the necessary variables |
| #5 / 28-04-2022 | Judge assigns ten (10) points to a team | The variable passes the exception handler and the points are added to the scoreboard |  | Program works ok, no problem found. |
| #6 / / 28-04-2022 | Judge assigns minus ten (-10) points to a team | The exception handler catches the error and throws an exception. No points are added to the scoreboard |  | Program works ok, no problem found. |

# **Conclusion and recommendations**

The scenario presented required of an application capable of manage user information for different competitions. Even as a barebone, the outcome of this project has been a solid base to build a complex data management application. Object Orientation allows developers to create well-structured and easy to maintain code, and it’s a very powerful tool that is a foundation for all the projects to come.

The chosen language for this project was Java: its main advantages is that is a simple language to use, with a clearer syntax than other languages, it is used for general purpose and can be run in all machines that support the language when compiled. However, it is still necessary to consider that it is slower than other compiled languages such as C++, and memory management is not very effective when working in larger applications.

Several good practices were used to ensure a correct application development, such as:

* Meaningful variable, class and method names, to clearly identify the elements that compose the application and what they do.
* Specific methods that execute a single function, avoiding long methods and high complexity levels.
* Proper comments throughout the application, explaining what is every step that is being executed.

For this application to really start to be functional, a method to handle user input should be implemented, as well as an object creation automation and organization. Also, it lacks several exception handlers for the object creation: exceptions to control the constructor for the classes in case the user is meant to input the information.

# **Appendix**

The Classes are ordered alphabetically:

1. Admin.java

package com.worldskillscompetition;  
  
public class Admin {  
  
 //Instance Variables  
 private static int *adminCounter* = 0;  
 private String aID;  
 private String username;  
 private String password;  
  
 //Constructor  
  
 public Admin( String username, String password) {  
 this.aID = String.*format*("%02d", ++*adminCounter*);  
 this.username = username;  
 this.password = password;  
 }  
  
 //Getters  
  
 public String getUsername() {  
 return username;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 //Setters  
  
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 // toString to show the class content  
 @Override  
 public String toString() {  
 return "Admin{" +  
 "aID='" + aID + '\'' +  
 ", username='" + username + '\'' +  
 ", password='" + password + '\'' +  
 '}';  
 }  
}

1. Category.java

package com.worldskillscompetition;  
  
public class Category {  
  
 //Instance Variables  
 private static int *categoryCounter* = 0;  
 private String catID;  
 private String catName;  
  
 //Constructor  
 public Category(String catName) {  
 this.catID = String.*format*("%02d", ++*categoryCounter*);  
 this.catName = catName;  
 }  
  
 //Getters  
  
 public String getCatID() {  
 return catID;  
 }  
  
 public String getCatName() {  
 return catName;  
 }  
  
 //Setters  
  
 public void setCatID(String catID) {  
 this.catID = catID;  
 }  
  
 public void setCatName(String catName) {  
 this.catName = catName;  
 }  
  
 // toString to show the class content  
 @Override  
 public String toString() {  
 return "Category{" +  
 "catID='" + catID + '\'' +  
 ", catName='" + catName + '\'' +  
 '}';  
 }  
}

1. College.java

package com.worldskillscompetition;  
  
public class College {  
  
 //Instance Variables  
 private static int *collegeCounter*;  
 private String cID;  
 private String cName;  
 private String city;  
 private String postcode;  
  
 //Constructor  
  
 public College(String cName, String city, String postcode) {  
 this.cID = String.*format*("%02d", ++*collegeCounter*);  
 this.cName = cName;  
 this.city = city;  
 this.postcode = postcode;  
 }  
  
 //Getters  
  
 public String getcID() {  
 return cID;  
 }  
  
 public String getcName() {  
 return cName;  
 }  
  
 public String getCity() {  
 return city;  
 }  
  
 public String getPostcode() {  
 return postcode;  
 }  
  
  
 //Setters  
  
 public void setcID(String cID) {  
 this.cID = cID;  
 }  
  
 public void setcName(String cName) {  
 this.cName = cName;  
 }  
  
 public void setCity(String city) {  
 this.city = city;  
 }  
  
 public void setPostcode(String postcode) {  
 this.postcode = postcode;  
 }  
  
 // toString to show the class content  
 @Override  
 public String toString() {  
 return "College{" +  
 "cID='" + cID + '\'' +  
 ", cName='" + cName + '\'' +  
 ", city='" + city + '\'' +  
 ", postcode='" + postcode + '\'' +  
 '}';  
 }  
}

1. ILadder.java

package com.worldskillscompetition;  
  
public interface ILadder {  
  
 //Methods  
 public int addPoints(int points);  
 public double addPoints(double points);  
 public int deletePoints(int points);  
 public double deletePoints(double points);  
}

1. IRole.java

package com.worldskillscompetition;  
  
public interface IRole {  
  
 //Methods  
 public String getRole();  
 public void setRole(String role);  
  
}

1. Main.Java

package com.worldskillscompetition;  
/\*  
Edinburgh college - Sighthill Campus  
H17135 - Software Development Object Oriented Programming  
Assesment 01  
Name: Adrian Sanchez Rodriguez  
ID: EC1939656  
Date: 28th April 2022  
Version: 1.0  
Notes: N/A  
 \*/  
  
import jdk.nashorn.internal.runtime.ECMAException;  
  
public class Main {  
 public static void main(String[] args){  
  
 //Instance variables. Used to check the admin credentials  
 String username = "Adrian";  
 String password = "Sanchez";  
  
 //Teacher roles  
 String[] teacherRoles = {"Teacher", "Coach", "Judge"};  
  
 //Score variables  
 String scoreBoard = "0.0";  
 double score = 0.0;  
 int points;  
 double pointsD;  
  
  
 //Adding admin to be able to navigate the app.  
 Admin admin = new Admin("Adrian", "Sanchez");  
  
 //Check administrator username and password  
 boolean userCheck = *checkCredentials*(username, password, admin);  
 //Check username  
 if (userCheck){  
 System.*out*.println("Login successful!");  
 }  
 else{  
 System.*out*.println("Invalid credentials");  
 }  
  
 //Create College objects  
 College college1 = new College("Milton Rd Campus", "Edinburgh", "EH15 2PQ");  
 System.*out*.println(college1.toString());  
 College college2 = new College("Sighthill Campus", "Edinburgh", "EH11 4DE");  
 System.*out*.println(college2.toString());  
  
 //Create student objects  
 Student student1 = new Student("student1","1234","Adrian Sanchez", "2/5 Bonnington Avenue", "07597129679","Full-Time");  
 System.*out*.println(student1.toString());  
 Student student2 = new Student("student2","1234","Mai Barbosa", "19 Raedburn pl", "07896234875","Part-Time");  
 System.*out*.println(student2.toString());  
 Student student3 = new Student("student3","1234","Iain Mckinnon", "2/5 Bonnington Avenue", "07896234875","Apprentice");  
 System.*out*.println(student3.toString());  
 Student student4 = new Student("student4","1234","Pav Gill", "85 Rose St", "07855324015","Full-Time");  
 System.*out*.println(student4.toString());  
 Student student5 = new Student("student5","1234","Bogdan Kaspersen", "22 Albert Road", "07597129679","Apprentice");  
 System.*out*.println(student5.toString());  
 Student student6 = new Student("student6","1234","Filibert Gold", "9729 King Street", "07896234875","Part-Time");  
 System.*out*.println(student6.toString());  
 Student student7 = new Student("student7","1234","Innocenzo Bonaccorso", "3 Park Lane", "07896234875","Part-Time");  
 System.*out*.println(student7.toString());  
 Student student8 = new Student("student8","1234","Quin Reece", "22 Springfield Road", "07855324015","Full-Time");  
 System.*out*.println(student8.toString());  
 Student student9 = new Student("student9","1234","Fridumar Moore", "90 Queen Street", "07597129679","Apprentice");  
 System.*out*.println(student9.toString());  
 Student student10 = new Student("student10","1234","Anantha Moretti ", "94 Mill Road", "07896234875","Part-Time");  
 System.*out*.println(student10.toString());  
 Student student11 = new Student("student11","1234","Svetlana Agricola ", "23 Grove Road", "07896234875","Part-Time");  
 System.*out*.println(student11.toString());  
 Student student12 = new Student("student12","1234","Fatimah Cullen ", "410 Stanley Road", "07855324015","Apprentice");  
 System.*out*.println(student12.toString());  
  
 //Create Teacher objects  
 Teacher teacher1 = new Teacher("teacher1","1234","Quin Reece", "22 Springfield Road", "07855324015");  
 teacher1.setRole(teacherRoles[0]);  
 System.*out*.println(teacher1.toString());  
 Teacher teacher2 = new Teacher("teacher2","1234","Iain Mckinnon","2/5 Bonnington Avenue","07597129679");  
 teacher2.setRole(teacherRoles[1]);  
 System.*out*.println(teacher2.toString());  
 Teacher teacher3 = new Teacher("teacher3","1234", "Mai Barbosa", "19 Raeburn Pl", "07856235486");  
 teacher3.setRole(teacherRoles[2]);  
 System.*out*.println(teacher3.toString());  
 Teacher teacher4 = new Teacher("teacher4", "1234", "Kumi Gilchrist", "85 Rose St", "07459654123");  
 teacher4.setRole(teacherRoles[1]);  
 System.*out*.println(teacher4.toString());  
 Teacher teacher5 = new Teacher("Teacher5", "1234", "Lorna Cuttle", "20 Shandwick Pl", "07419658423");  
 teacher5.setRole(teacherRoles[1]);  
 System.*out*.println(teacher5.toString());  
  
 //Create Competition category objects  
 Category category1 = new Category("Software Development");  
 System.*out*.println(category1.toString());  
 Category category2 = new Category("Networking");  
 System.*out*.println(category2.toString());  
 Category category3 = new Category("Web Design");  
 System.*out*.println(category3.toString());  
  
 //Create Team objects  
 Team team1 = new Team("Team1",category1,college1,teacher2,student1,student2,student3,student4);  
 System.*out*.println(team1.toString());  
 Team team2 = new Team("Team2",category2,college1,teacher4,student5,student6,student7,student8);  
 System.*out*.println(team2.toString());  
 Team team3 = new Team("Team3",category3,college2,teacher5,student9,student10,student11,student12);  
 System.*out*.println(team3.toString());  
  
 //Judge Scoring Team 1 (integer points)  
 points = 10;  
 //Check for negative values in points  
 try{  
 *checkPointInput*(points);  
 }  
 catch(Exception e){  
 System.*out*.println("The system can't accept negative values");  
 points = 0;  
 }  
  
 score += teacher3.addPoints(points);  
 scoreBoard = Double.*toString*(score);  
 team1.setTeamScore(scoreBoard);  
 System.*out*.println("Judge " + teacher3.getName() + " scored " + points + " points to " + team1.getTeamName() + ".");  
 System.*out*.println(team1.toString());  
  
 //Judge Scoring Team2 (double points)  
 score = 0;  
 pointsD = 12.7;  
 //Check for negative values in points  
 try{  
 *checkPointInput*(pointsD);  
 }  
 catch(Exception e){  
 System.*out*.println("The system can't accept negative values");  
 pointsD = 0;  
 }  
  
 score += teacher3.addPoints(pointsD);  
 scoreBoard = Double.*toString*(score);  
 team2.setTeamScore(scoreBoard);  
 System.*out*.println("Judge " + teacher3.getName() + " scored " + pointsD + " points to " + team2.getTeamName() + ".");  
 System.*out*.println(team2.toString());  
 }  
 private static void checkPointInput(int points) {  
 if (points < 0 ){  
 throw new ArithmeticException("The system can't accept negative values");  
 }  
 }  
 private static void checkPointInput(double points) {  
 if (points < 0.0 ){  
 throw new ArithmeticException("The system can't accept negative values");  
 }  
 }  
  
 private static boolean checkCredentials(String username, String password, Admin admin) {  
 return(admin.getUsername().equals(username)) && (admin.getPassword().equals(password));  
 }  
}

1. Person.java

package com.worldskillscompetition;  
  
public abstract class Person {  
  
 //Instance Variables  
 protected static int *personCounter* = 0;  
 protected String pID;  
 protected String username;  
 protected String password;  
 protected String name;  
 protected String address;  
 protected String telephone;  
  
 //Constructor  
 public Person(String username, String password, String name, String address, String telephone) {  
 this.pID = String.*format*("%02d", ++*personCounter*);  
 this.username = username;  
 this.password = password;  
 this.name = name;  
 this.address = address;  
 this.telephone = telephone;  
 }  
  
 //Getters  
  
 public abstract String getpID();  
 public abstract String getUsername();  
 public abstract String getPassword();  
 public abstract String getName();  
 public abstract String getAddress();  
 public abstract String getTelephone();  
  
 //Setter  
  
 public abstract void setpID(String pID);  
 public abstract void setUsername(String username);  
 public abstract void setPassword(String password);  
 public abstract void setName(String name);  
 public abstract void setAddress(String address);  
 public abstract void setTelephone(String telephone);  
}

1. Student.java

package com.worldskillscompetition;  
  
public class Student extends Person{  
  
 //Instance variables  
 private String status;  
  
 //Constructor  
 public Student(String username, String password, String name, String address, String telephone, String status) {  
 super(username, password, name, address, telephone);  
 this.status = status;  
 }  
  
 //parent getters  
 @Override  
 public String getpID() {  
 return pID;  
 }  
  
 @Override  
 public String getUsername() {  
 return username;  
 }  
  
 @Override  
 public String getPassword() {  
 return password;  
 }  
  
 @Override  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public String getAddress() {  
 return address;  
 }  
  
 @Override  
 public String getTelephone() {  
 return telephone;  
 }  
  
 //Child Getters  
 public String getStatus() {  
 return status;  
 }  
  
  
 //parent setters  
  
 @Override  
 public void setpID(String pID) {  
 this.pID = pID;  
 }  
  
 @Override  
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
 @Override  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 @Override  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 @Override  
 public void setTelephone(String telephone) {  
 this.telephone = telephone;  
 }  
  
 //Child Setters  
  
 public void setStatus(String status) {  
 this.status = status;  
 }  
  
 // toString to show the class content  
  
 @Override  
 public String toString() {  
 return "Student{" +  
 "pID='" + pID + '\'' +  
 ", username='" + username + '\'' +  
 ", password='" + password + '\'' +  
 ", name='" + name + '\'' +  
 ", address='" + address + '\'' +  
 ", telephone='" + telephone + '\'' +  
 " status='" + status + '\'' +  
 '}';  
 }  
}

1. Teacher.java

package com.worldskillscompetition;  
  
public class Teacher extends Person implements ILadder,IRole{  
  
 //Instance variables  
 private String tRole;  
 private int tScore;  
 private double tScoreD;  
  
 //Constructor  
 public Teacher(String username, String password, String name, String address, String telephone) {  
 super(username, password, name, address, telephone);  
 this.tScore = 0;  
 this.tScoreD = 0.0;  
 }  
  
 //parent getters  
  
 @Override  
 public String getpID() {  
 return pID;  
 }  
  
 @Override  
 public String getUsername() {  
 return username;  
 }  
  
 @Override  
 public String getPassword() {  
 return password;  
 }  
  
 @Override  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public String getAddress() {  
 return address;  
 }  
  
 @Override  
 public String getTelephone() {  
 return telephone;  
 }  
  
 //Parent setters  
  
 @Override  
 public void setpID(String pID) {  
 this.pID = pID;  
 }  
  
 @Override  
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
 @Override  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 @Override  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 @Override  
 public void setTelephone(String telephone) {  
 this.telephone = telephone;  
 }  
  
  
 //ILadder Methods  
 @Override  
 public int addPoints(int points) {  
 return this.tScore = points;  
 }  
  
 @Override  
 public double addPoints(double points) {  
 return this.tScoreD = points;  
 }  
  
 @Override  
 public int deletePoints(int points) {  
 return tScore = points;  
 }  
  
 @Override  
 public double deletePoints(double points) {  
 return tScoreD = points;  
 }  
  
 //IRole Methods  
 @Override  
 public String getRole() {  
 return tRole;  
 }  
  
 @Override  
 public void setRole(String role) {  
 this.tRole = role;  
 }  
  
 // toString to show the class content  
 @Override  
 public String toString() {  
 return "Teacher{" +  
 "pID='" + pID + '\'' +  
 ", username='" + username + '\'' +  
 ", password='" + password + '\'' +  
 ", name='" + name + '\'' +  
 ", address='" + address + '\'' +  
 ", telephone='" + telephone + '\'' +  
 " role='" + tRole + '\'' +  
 '}';  
 }  
}

1. Team.java

package com.worldskillscompetition;  
  
public class Team {  
  
 //Instance Variables  
 private static int *teamCounter*;  
 private String teamID;  
  
 private String teamName;  
 private String category;  
 private String college;  
 private String teacher;  
 private String member1, member2, member3, member4;  
 private String teamScore;  
  
 //Constructor  
  
 public Team(String teamName, Category category, College college, Teacher teacher, Student member1, Student member2, Student member3, Student member4) {  
 this.teamID = String.*format*("%02d", ++*teamCounter*);  
 this.teamName = teamName;  
 this.category = category.getCatName();  
 this.college = college.getcName();  
 this.teacher = teacher.getName();  
 this.member1 = member1.getName();  
 this.member2 = member2.getName();  
 this.member3 = member3.getName();  
 this.member4 = member4.getName();  
 this.teamScore = "0.0";  
 }  
  
 //Getters  
  
 public String getTeamID() {  
 return teamID;  
 }  
  
 public String getTeamName() {  
 return teamName;  
 }  
  
 public String getCategory() {  
 return category;  
 }  
  
 public String getCollege() {  
 return college;  
 }  
  
 public String getTeacher() {  
 return teacher;  
 }  
  
 public String getMember1() {  
 return member1;  
 }  
  
 public String getMember2() {  
 return member2;  
 }  
  
 public String getMember3() {  
 return member3;  
 }  
  
 public String getMember4() {  
 return member4;  
 }  
  
 public String getTeamScore() {  
 return teamScore;  
 }  
  
 //Setters  
  
 public static void setTeamCounter(int teamCounter) {  
 Team.*teamCounter* = teamCounter;  
 }  
  
 public void setTeamID(String teamID) {  
 this.teamID = teamID;  
 }  
  
 public void setTeamName(String teamName) {  
 this.teamName = teamName;  
 }  
  
 public void setCategory(String category) {  
 this.category = category;  
 }  
  
 public void setCollege(String college) {  
 this.college = college;  
 }  
  
 public void setTeacher(String teacher) {  
 this.teacher = teacher;  
 }  
  
 public void setMember1(String member1) {  
 this.member1 = member1;  
 }  
  
 public void setMember2(String member2) {  
 this.member2 = member2;  
 }  
  
 public void setMember3(String member3) {  
 this.member3 = member3;  
 }  
  
 public void setMember4(String member4) {  
 this.member4 = member4;  
 }  
  
 public void setTeamScore(String teamScore) {  
 this.teamScore = teamScore;  
 }  
  
 // toString to show the class content  
 @Override  
 public String toString() {  
 return "Team{" +  
 "teamID='" + teamID + '\'' +  
 ", category='" + category + '\'' +  
 ", college='" + college + '\'' +  
 ", teacher='" + teacher + '\'' +  
 ", member1='" + member1 + '\'' +  
 ", member2='" + member2 + '\'' +  
 ", member3='" + member3 + '\'' +  
 ", member4='" + member4 + '\'' +  
 ", teamScore='" + teamScore + '\'' +  
 '}';  
 }  
}