

**Software Development Report**  
for  
**CSC 122: Programming II: Fall 2022**  
**Own Data Structure:**  
**Set**  
by  
**Manel Casado**

**Problem Summary**

The program allows to store swimming teams with their code and score and their swimmers and do actions with them like as an example showing a list of all the teams, adding a team, add a swimmer to a team (indicate an error if it is already exists), show a list of all the swimmers a team has...

**Implementation Requirements**

- Set
- Independent learning
- @override
- Dates
- Parse

## System Design

### SwimmingPlayer

```
-Set<Team>teams

+SwimmingPlayer()
+main(String[] args)
-showOptions()
-showAllTeams(sortField : String)
-addTeam(team : Team)
-removeTeam(team : Team)
-addMember(team : Team, person : Person)
-removeMember(team : Team, person : Person)
-findTeam(team : Team) : Team
-showMembers(team : Team)
```

### Team

```
-code : String
-name : String
-score : int
-persons : Set<Persons>

+Team(code : String, name : String, score : int)
+addPerson(aPerson : Person)
+removePerson(aPerson : Person)
+compareTo(otherTeam : Team) : int
+hashCode() : int
+equals(obj : Object) : boolean
+getMembersOrderedBy(sortField : String)
+toString() : String
```

### Person

```
+MALE = "Male"
+FEMALE = "Female"
-id : int
-name : String
-lastName : String
-birthday : Date
-gender : String

+Person(id : int, name : String, lastName : String,
birthday : Date, gender : String)
+hasCode() : int
+equals(obj : Object) : boolean
+compareTo(otherPerson : Person) : int
+equals(otherPerson : Person) : boolean
+toString() : String
```

## Testing Plan

Cas e	Description	Input	Output
S1	Create a team	<pre>aTeam = new Team("BU-IN", "Bethel University (IN)", 0); addTeam(aTeam);</pre>	Team added. Current teams are: Code= BU-IN, name:Bethel University (IN), score:0, swimmers: null
S2	Create a swimmer	<pre>birthday = null; try {     birthday =new SimpleDateFormat("MM/dd/yyyy").parse("05/12/2003"); } catch (ParseException e) {     System.out.println("Invalid date"); } aSwimmer = new Person(1, "Manel", "Casado Garrigues", birthday, Person.MALE);</pre>	Swimming created

S3	Add swimmer to a team	<pre> aTeam = new Team("BU-IN", "Bethel University (IN)", 0); try {     swimingPlayer.addMember(aTeam, aSwimmer); } catch (Exception e) {     System.err.println(e.getLocalizedMessage() + " " + aTeam); } </pre>	Person added. [Code= BU-IN, name: Bethel University (IN), score: 0, swimmers: [Person with id:= 1, name: Manel, lastName: Casado Garrigues, birthDay: Mon May 12 00:00:00 GMT 2003, gender: Male]
S4	Remove a team	<pre> aTeam = new Team("BU-IN", "Bethel University (IN)", 0); swimingPlayer.removeTeam(aTeam); </pre>	Team does not exist Code= BU-IN, name: Bethel University (IN), score: 0, swimmers: null
S5	Sort team members	Not done	Not done

### Time Spent

The program and development report took me around 10 hours to develop

### Outside resources used

- UML cheat sheet provided in Canvas
- Microsoft Visio to create the System Design
- Link with examples of data structures and how they work provided on the syllabus
- [geeksforgeeks.org](http://geeksforgeeks.org) for understanding how Set works and help from my dad creating a testing plan

### Security Report

- Positive: There cannot be two swimmers with the same name
- Negative: Not being able to read

### Ethical Report

- A potential ethical risk would be if the data got leaked and contained sensible information in a more advanced version
- A positive application would be applying this to an actual swimming league. This could be used by swimming associations in a more developed version.

**Future improvements**

- Load data from a file or through the console instead of adding it on the code
- Throw exception with an error if a team already exists
- Order team members by different criteria
- Add a menu with different options such as add, remove, etc.

**Lessons learned**

- How Sets work and how to use them
- What @override is

**Improvements of Work**

- Better testing