

LAB 1: A Java scoreboard

Getting started

In this lab you will examine and modify Java classes which represent a scoreboard at a football match.

Task 1 : Examine the classes

Writing Java code:

- 1. Download and open the BlueJ project *lab1*. Note that there are two classes.
- 2. Open each class in the BlueJ editor and carefully read through the code.
- 3. Go to the Assignments section for this module on Blackboard. The assignment *Lab 1 Questions* consists of a set of short-answer and multiple choice questions based on the code in the *lab1* project. Start the assignment and attempt all questions. You will probably need to refer back to the code in order to answer these.

Task 2 : Test the classes

You will now test the classes in the *lab1* project.

Testing the TeamScore class:

Test the TeamScore class as follows.

1. Right-click on the TeamScore class in BlueJ and select the constructor

TeamSco	re
	new TeamScore(String teamName)
	Open Editor
	Compile
	Inspect
	Remove
	Create Test Class



2. Enter a value for teamName in the Create Object dialog (choose any team name you like)

🚳 BlueJ: Create Object 🛛 🔀					
<pre>// constructor for objects of type TeamScore // @param teamName the team name TeamScore(String teamName)</pre>					
Name of Instance: teamScor1					
new TeamScore ("Inter Milan"					
Ok Cancel					

3. Check that an object is created in the Object Bench

	teamScort: TeamScore
C	reating object Done.

4. Right click on the object and call getTeamName. Check that the team name you entered previously is returned

			🧆 BlueJ:	Meth	od Result	
En	inherited from Object		// returns //	the tea	am name	
Can	int getScore()		// @returi	n the te	am name	
	String getTeamName()		String get	reamNa	me()	
	void resetScore() void setTeamName(String teamName)	-	teamSco returne	or1.getT ed:	eamName()	Inspect
TeamSi	void updateScore()		String		"Inter Milan"	
	Inspect Remove					Close

5. Similarly, call getScore. Check that a value of 0 is returned



M1G413283: Introduction to Programming 2

🤏 BlueJ: Meth							
// returns the score for the team //							
int getScore()	// @return_the_score int_getScore() 						
teamScor1.getS	icore()	Inspect					
int	Get						
		Close					

- 6. Call updateScore twice. Now call getScore and check that a value of **2** is returned
- 7. Call resetScore. Now call getScore and check that a value of **0** is returned
- 8. Call setTeam name and enter a different team name in the Method Call dialog



9. Call getTeamName. Check that the team name you entered previously is returned

Testing the Scoreboard class:

Test the ScoreBoard class as follows.

- 1. Right-click on the ScoreBoard class in BlueJ and select the constructor
- 2. The constructor takes no parameters, so you don't need to enter any values in the Create Object dialog. Just click OK and check that an object is created in the Object Bench.



🌁 BlueJ: Create (Object 🛛 🛛
// constructor for o ScoreBoard()	bjects of type ScoreBoard
Name of Instance:	scoreBoal
	Ok Cancel

 Right-click on the object, and choose Inspect from the menu. Check that the ScoreBoard object has two fields, home and away, which contain references to TeamScore objects, and an integer field. Select the home field and click Inspect – check that you then see a representation of an object of type TeamScore.

		-A	BlueJ: Object Inspector	
BlueJ: Object Inspector <u>scoreBoa1 : ScoreBoard</u> private TeamScore home Get	-		home : TeamScore private String teamName "HOME" private int score 0	Inspect Get
private TeamScore away private int minutesPlayed 0		l	Show static fields	Close
Show static fields Close	J			

- 4. Right click on the object and call setMinutesPlayed. Enter a value of **45** in the Method Call dialog. Now call setMinutesPlayed and check that the value of 45 is returned.
- 5. Create a new TeamScore object in the object bench with a team name of "Juventus". Note the name of the object in the Object Bench – probably teamScor1.
- 6. Right click on the object and call setHome. Enter the name of the TeamScore object in the Method Call dialog (or you can simply click in the text box and then click on the TeamScore object in the Object Bench).



M1G413283: Introduction to Programming 2

Status		🚳 BlueJ: Method Call	×
Run Tests recording End		// set the home team score // // @param home the home team score void setHome(TeamScore home)	_
Cancel		scoreBoa1.setHome(<u>teamScor1</u>))	
scoreBoa1:	team	Ok Cancel	
ScoreBoard	Team	Score	

7. Inspect the ScoreBoard object and in turn inspect the home field. Check that the team name is "Juventus".

	_	🚳 BlueJ: Object Inspector	
BlueJ: Object Inspector		home : TeamScore private String teamName "Juventus" private int score 0	Inspect Get
private TeamScore away of Get private int minutesPlayed 0 Show static fields Close		Show static fields	Close
	ļ		



Task 3: Adding functionality to the ScoreBoard class

Writing Java code:

- 1. Open the ScoreBoard class in the BlueJ editor and edit the Java code so that your class has the following additional methods:
 - A method called goalScored, with no return value, which takes a single String parameter and does the following:
 - if the parameter value is "h", calls the updateScore method of the home object
 - else if the parameter value is "a", calls the updateScore method of the away object
 - A method called display, which displays the current score in the match, using the following code:

```
public void display()
{
    System.out.format("%d minutes played\n", minutesPlayed);
    System.out.format("%s %d:%d %s\n\n",
        home.getTeamName(),home.getScore(),
        away.getScore(),away.getTeamName());
}
```

- A method called setUpGame, which takes two String parameters and has no return value, which does the following:
 - o Sets the value of minutesPlayed to 0
 - Calls home.setTeamName to set the name of the home team to the first parameter value
 - o Calls home.resetScore to set the home team score to 0
 - Calls away.setTeamName to set the name of the away team to the first parameter value
 - o Calls away.resetScore to set the away team score to 0

Testing:

Test the ScoreBoard class as follows. If a test doesn't work, go back to your code and try to identify and fix the problem.



- 1. Right-click on the ScoreBoard class in BlueJ and call the constructor. Check that an object is created in the Object bench.
- 2. Right click on the object and call display. The Terminal Window should open and you should see the following output:



3. Right click on the object and call setUpgame. Enter two team names in the Method Call dialog box.

🚳 BlueJ: Method Call	×
// set up the teams for a new game // // @param homeTeam the home team name // @param awayTeam the away team name void setUpGame(String homeTeam, String awayTeam)	
scoreBoa1.setUpGame (<mark>"Inter Milan" v</mark> , String homeTeam "Roma" v) String awayTeam	
Ok Cancel)

4. Right click on the object and call goalScored. Enter "h" in the Method Call dialog box.

🚳 BlueJ: Method Call	×
// updates the score when a goal is scored // // @param team /void goalScored(String team)	
scoreBoa1.goalScored (📷 💌)	
Ok Cancel	

5. Right click on the object and call the goalScored method again with the same parameter.



- 6. Right click on the object and call the goalScored method again and enter "a" in the Method Call dialog.
- 7. Right click on the object and call the setMinutesPlayed method. Enter **45** in the Method Call dialog box.



8. Right click on the object and call display. The Terminal Window should now show the following output:

🚳 BlueJ: Terminal Window - lab1	
Options	
0 minutes played	^
HOME 0:0 AWAY	
45 minutes played	
Inter Milan 2:1 Roma	
	v
	>