

Past-paper walk-through

Methods: How to Write Them ■ 3.5

eXercise

Questions in this set

- 2024 Q2
- 2023 Q1
- 2022 Q2
- 2019 Q1
- 2017 Q2
- 2016 Q2
- 2016 Q4
- 2015 Q4

Reading the mark scheme

- **B1** a mark that stands on its own – the point is either made or not.
- **M1** a *method* mark: the right approach, even if the number is wrong.
- **A1** an *answer* mark, and it usually needs its M mark first.
- **C1** a working mark on the way to the answer.
- **//** separates answers the examiner accepts equally.
- **ecf** = error carried forward: a later part is marked on your own earlier answer.
- **owtte** = “or words to that effect” – the wording need not match.

Question 2

2024 ■ Q2

This question involves a scoreboard for a game. The game is played between two teams who alternate turns so that at any given time, one team is active and the other is inactive. During a turn, a team makes one or more plays. Each play can score one or more points and the team's turn continues, or the play can fail, in which case no points are scored and the team's turn ends. The Scoreboard class, which you will write, is used to keep track of the score in a game.

The Scoreboard class contains a constructor and two methods.

Question 2

- The constructor has two parameters. The first parameter is a `String` containing the name of team 1, and the second parameter is a `String` containing the name of team 2. The game always begins with team 1 as the active team.
- The `recordPlay` method has a single nonnegative integer parameter that is equal to the number of points scored on a play or 0 if the play failed. If the play results in one or more points scored, the active team's score is updated and that team remains active. If the value of the parameter is 0, the active team's turn ends and the inactive team becomes the active team. The `recordPlay` method does not return a value.
- The `getScore` method has no parameters. The method returns a `String` containing information about the current state of the game. The returned string begins with the score of team 1, followed by a hyphen ("-"), followed by the score of team 2, followed by a hyphen, followed by the name of the team that is currently active.

Question 2

The following table contains a sample code execution sequence and the corresponding results. The code execution sequence appears in a class other than Scoreboard.

Statement	Value Returned (blank if none)	Explanation
String info; Scoreboard game = new Scoreboard("Red", "Blue"); info = game.getScore(); game.recordPlay(1);	"0-0-Red"	game is a new Scoreboard for a game played between team 1, whose name is "Red", and team 2, whose name is "Blue". The active team is set to team 1.
info = game.getScore(); game.recordPlay(0); info = game.getScore();	"1-0-Red"	Team 1 earns 1 point because the game always begins with team 1 as the active team.
info = game.getScore(); game.recordPlay(3); info = game.getScore();	"1-0-Blue"	Team 1's play failed, so team 2 is now active.
info = game.getScore(); game.recordPlay(3); info = game.getScore();	"1-0-Blue"	The score and state of the game are unchanged since the last call to getScore.
info = game.getScore(); game.recordPlay(1); game.recordPlay(0); info = game.getScore();	"1-3-Blue"	Team 2 earns 3 points.
info = game.getScore(); game.recordPlay(0); game.recordPlay(4); game.recordPlay(0); info = game.getScore();	"1-3-Blue"	Team 2 earns 1 point.
info = game.getScore(); game.recordPlay(0); game.recordPlay(4); game.recordPlay(0); info = game.getScore();	"1-4-Red"	Team 2's play failed, so team 1 is now active.
info = game.getScore(); game.recordPlay(0); game.recordPlay(4); game.recordPlay(0); info = game.getScore();	"1-4-Red"	Team 1's play failed, so team 2 is now active.
info = game.getScore(); game.recordPlay(0); game.recordPlay(4); game.recordPlay(0); info = game.getScore();	"1-8-Red"	Team 2 earns 4 points.
info = game.getScore(); game.recordPlay(0); game.recordPlay(4); game.recordPlay(0); info = game.getScore();	"1-8-Red"	Team 2's play failed, so team 1 is now active.
Scoreboard match = new Scoreboard("Lions", "Tigers"); info = match.getScore(); info = game.getScore();	"0-0-Lions"	match is a new and independent Scoreboard object.
info = game.getScore();	"1-8-Red"	

Question 2

Write the complete Scoreboard class. Your implementation must meet all specifications and conform to the examples shown in the preceding table.

Question 2

<code>info = game.getScore();</code>	<code>"1-0-Red"</code>	
<code>game.recordPlay(0);</code>		Team 1's play failed, so team 2 is now active.
<code>info = game.getScore();</code>	<code>"1-0-Blue"</code>	
<code>info = game.getScore();</code>	<code>"1-0-Blue"</code>	The score and state of the game are unchanged since the last call to <code>getScore</code> .
<code>game.recordPlay(3);</code>		Team 2 earns 3 points.
<code>info = game.getScore();</code>	<code>"1-3-Blue"</code>	
<code>game.recordPlay(1);</code>		Team 2 earns 1 point.
<code>game.recordPlay(0);</code>		Team 2's play failed, so team 1 is now active.
<code>info = game.getScore();</code>	<code>"1-4-Red"</code>	
<code>game.recordPlay(0);</code>		Team 1's play failed, so team 2 is now active.
<code>game.recordPlay(4);</code>		Team 2 earns 4 points.
<code>game.recordPlay(0);</code>		Team 2's play failed, so team 1 is now active.
<code>info = game.getScore();</code>	<code>"1-8-Red"</code>	
<code>Scoreboard match = new Scoreboard("Lions", "Tigers");</code>		match is a new and independent Scoreboard object.
<code>info = match.getScore();</code>	<code>"0-0-Lions"</code>	
<code>info = game.getScore();</code>	<code>"1-8-Red"</code>	

Question 1

2023 ■ Q1

This question involves the `AppointmentBook` class, which provides methods for students to schedule appointments with their teacher. Appointments can be scheduled during one of eight class periods during the school day, numbered 1 through 8. A requested appointment has a duration, which is the number of minutes the appointment will last. The 60 minutes within a period are numbered 0 through 59. In order for an appointment to be scheduled, the teacher must have a block of consecutive, available minutes that contains at least the requested number of minutes in a requested period. Scheduled appointments must start and end within the same period.