

Date.java

Test Case	Description	Sample Input	Expected result/output
1	Testing the Date(String date) constructor	"05/32/1999"	32 is not a valid day.
2	Testing the Date(Date date) constructor and us toString() to display the contents of the object	Date test = new Date("12/13/2017"); Date test2 = new Date(test);	12/13/2017
3	Testing the isValid() function and using the toString() method on leap year dates and regular months.	Case 1: "5/32/1999" Case 2: "5/9/1999" Case 3: "2/29/2012" Case 4: "2/29/2013"	Case 1: Date : 5/32/1999 isValid() = false 32 is not a valid day. Case 2: Date : 5/9/1999 isValid() = true Case 3: Date : 2/29/2012 isValid() = true Case 4: Date : 2/29/2013 isValid() = false
4	Testing isLeapYear()	Case 1: 02/29/2012 Case 2: 02/29/2013	Case 1: true Case 2: false
5	Testing validateMonth()	Case 1: 02/29/2013 Case 2: 15/29/2013	Case 1: true Case 2: false

6	Testing validateDay()	Case 1: 05/09/1999 Case 2: 05/32/1999	Case 1: true Case 2: false
---	-----------------------	--	-------------------------------------

TeamMember.java

Test Case	Description	Sample Input	Expected result/output
1	Testing the TeamMember() constructor and using toString() to display the contents	Date testDate2 = new Date("05/09/1999") "Raina", testDate2	Raina 5/9/1999
2	Testing the equals() method and displaying the results	COMPARE WITH: Raina 5/9/1999 Case 1: Raina 8/13/13 Case 2: Raina 5/9/1999	Case 1: false Case 2: true
3	Testing getStartDate() method on a TeamMember object and displaying the Date with the toString() method.	TeamMember jay = new TeamMember("Jay", new Date("12/1/2014")); jay.getStartDate();	12/1/2014

Team.java

Test Case	Description	Sample Input	Expected result/output
1	Testing Team() constructor, adding a TeamMember to the team	<pre>TeamMember jay = new TeamMember("Jay", new Date("12/1/2014")); Team testTeam = new Team(); testTeam.add(jay);</pre>	Jay has joined the team!
2	Adding a TeamMember that has already been added by invoking the contains() function	<pre>testTeam.add(bob); testTeam.add(bob);</pre>	This member already exists!
3	Removing a TeamMember that does not exist	<pre>TeamMember al = new TeamMember("Al", new Date("2/29/2012")); testTeam.remove(al);</pre>	Al 2/29/2012 is not a team member.
4	Removing a TeamMember that does exist and invoking the find() function to find the target TeamMember	<pre>testTeam.add(al); testTeam.remove(al);</pre>	<p>Al 2/29/2012 has joined the team!</p> <p>Al 2/29/2012 has left the team!</p>
5	Adding more than GROW_SIZE (4) number of team members, to invoke the grow() function, also	<pre>testTeam.add(jay); testTeam.add(al); testTeam.add(jenny); testTeam.add(test);</pre>	<p>Jay 12/1/2014 has joined the team!</p> <p>Al 2/29/2012</p>

	adding a member with a repeated name, but different date	testTeam.add(compare1);	has joined the team! Jenny 6/30/2005 has joined the team! Raina 5/9/1999 has joined the team! Raina 8/13/13 has joined the team!
--	--	-----------------------------	---