Date Class: IsValid() method

Java class/method name being tested: Date class, isValid() method			
Test Case #	**Requirement**	**Test description and Input Data**	**Expected result/output**
1	The method shall not accept any date with the year before 1900.	• Create an instance of Date with valid day and month but with the year < 1900. • test data: "11/21/800"	false
2	Number of days in February for a non leap year shall be 28.	• Create an instance of Date with the month = 2, day > 28, and the year is a non-leap year • test data: "2/29/2018	false
3	Number of days in February for a leap year shall be 29.	• Create an instance of Date with the month = 2, day = 29, and the year is a leap year • test data: "2/29/2018	true
4	Valid range for the month shall be 1-12	• Create an instance of Date with valid day and year but with month > 13 • test data: "13/21/1999"	false

Member Class: CompareTo() method

Java class/method name being tested: Member class, CompareTo() method			
Test Case #	**Requirement**	**Test description and Input Data**	**Expected result/output**
1	Last name of m1 alphabetically less than m2 m1.compareTo(m2)	• Create an instance of member1 and member 2 with last name of m1 alphabetically before last name of m2 • test data:	-1

		Member m1 = new Member("Leah", "Ranavat",d1,e1,PISCATAWAY); Member m2 = new Member("Tanvi", "Thigle",d2,e2, BRIDGEWATER);	
2	Same members but different order of calling the method m2.compareTo(m1)	• Since order is different, will return positive one instead of negative	1
3	If last names are the same, compare to the first name	• Create an instance of member2 and member 3 with last name of m1 alphabetically before last name of m2 • test data: Member m2 = new Member("Tanvi", "Thigle",d2,e2, BRIDGEWATER); Member m3 = new Member("Mugdha","Thigle",d2,e2, FRANKLIN);	1
4	If the last name and first names are same should return 0	• test data: Member m2 = new Member("Tanvi", "Thigle",d2,e2, BRIDGEWATER); Member m4 = new Member("Tanvi","Thigle",d2,e2,FR ANKLIN);	0