Java class: `public class Date implements Comparable<Date> {}`			

List of the constructors:
`public Date(String date) {} /* Takes a date in "mm/dd/yyy" String format and converts it into a Date object */`
`Public Date() {} // Creates a Date object with today's date`			

Method signature: `public boolean isValid() {}`			
Test Case #	**Requirement**	**Test Description and Input Data**	**Expected Result / Output**
1	Do not accept dates earlier than the 1980s.	- Create Date instance with valid day and month, but year before the 1980s - Test Data: “10/20/1979”	false
2	Do not accept dates later than today’s date.	- Create Date instance with valid day and month, but year after 2021 - Test Data: “10/20/2022”	false
3	The entered month cannot be before January or after December (input string must have month value between 1 and 12).	- Create Date instances with valid days and years, but month values of 0 and 13 - Test Data: 1. “0/20/2020” 2. “13/20/2020”	1. false 2. false
4	For January, March, May, July, August, October, and December, the month can have anywhere between 1 and 31 days.	- Create Date instance with valid year and listed month value, but test the day values - Test Data: 1. “1/32/2020” 2. “1/0/2020” 3. “1/31/2020”	1. false 2. false 3. true
5	For April, June, September, and November, the month can have anywhere between 1 and 30 days.	- Create Date instance with valid year and listed month value, but test the day values - Test Data: 1. “4/31/2020” 2. “4/0/2020” 3. “4/30/2020”	1. false 2. false 3. true
6	For leap years, February can have between 1 and	- Create Date instance of a leap year with a valid month value and test	1. true 2. false

	29 days.	the day values - Test Data: 1. “2/29/2020” 2. “2/30/2020”	
7	For non-leap years, February can have between 1 and 28 days.	- Create Date instance of a leap year with a valid month value and test the day values - Test Data: 1. “2/28/2019” 2. “2/29/2019”	1. true 2. false