

Class name: Date

Method signature: public boolean isValid() {} //check if given date is a valid calendar date

Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method will return false for any February 29th that is in a non-leap year.	<ul style="list-style-type: none">Create an instance of Date with February as the month and a non-leap year as the year but not a valid day since the number of days in the month does not go up to 29.Test input: "02/29/2015"	false
2	This method shall return false for 00 entered as the day.	<ul style="list-style-type: none">Create an instance of Date with February as the month and a non-leap year as the year but not a valid day since the number of days in the month does not go up to 29.Test input: "02/00/2020"	false
3	This method shall return false for any month entered which is greater than 12.	<ul style="list-style-type: none">Create an instance of Date with a valid day and year but with a month that is greater than 12.Test input: "13/01/2017"	false
4	This method shall return false for 0000 entered as the year.	<ul style="list-style-type: none">Create an instance of Date with a valid day and month but with a 0000 year.Test input: "01/01/0000"	false
5	This method shall return false for a year which is in the future.	<ul style="list-style-type: none">Create an instance of Date with a valid day and month but with a year that is in the future.Test input: "02/20/3000"	false
6	This method shall return true for any February 29th that is in a leap year.	<ul style="list-style-type: none">Create an instance of Date with February as the month and a leap year as the year and valid day since the number of days in the month includes 29.Test input: "02/29/2016"	true
7	This method shall return true for any month which has 31 days in it.	<ul style="list-style-type: none">Create an instance of Date with the day as 31 and any Month and year.Test input: "01/31/2000"	True

Class name: Student

Method signature: public int compareTo() {} //return int when comparing students

Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method will return -1 for any student whose last name comes first in the alphabet.	<ul style="list-style-type: none">Create an instance of Student in which the student being compared to has a last name that comes first in alphabetical order.Test input: "Carl Brown 10/7/2004" "Kate Lindsey 7/15/2002"	-1
2	This method shall return -1 for any student whose last names are the same but whose first name comes first in the alphabet.	<ul style="list-style-type: none">Create an instance of Student in which the student being compared has the same last name but their first name comes first alphabetically.Test input: "April Doe 1/20/2003" "john Doe 1/20/2003"	-1
3	This method shall return 0 if the two students you compare are the same.	<ul style="list-style-type: none">Create an instance of Student in which the student being compared has the same last name, first name, and date of birth.Test input: "Jane Doe 5/1/1996" "Jane Doe 5/1/1996"	0
4	This method will return 1 for any student whose last name comes after in the alphabet.	<ul style="list-style-type: none">Create an instance of Student in which the student being compared has a last name that comes later in comparison alphabetically.Test input: "Mary Lindsey 12/1/2001" "Roy Brooks 8/8/1999"	1
5	This method shall return 1 for any student whose last names are the same but whose first name comes after in the alphabet.	<ul style="list-style-type: none">Create an instance of Student in which the student being compared has the same last name but their first name comes later in comparison alphabetically.Test input: "Jane Doe 5/1/1996" "April Doe 1/20/2003"	1