**Date.isValid() Test Cases**

| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| --- | --- | --- | --- |
| 1 | * Valid range for month is 1-12 | * Create instance of Date class that has month outside of valid range * Test data: “15/21/2022” | false |
| 2 | * Valid range for day for Jan, March, May, July, August, October and December is 1-31 | * Create instance of Date class that has day out of range * Test data: “1/41/2013” | false |
| 3 | * Valid range for day for April, June, September and November, is 30 days | * Create instance of Date class that has day out of range * Test data: “6/31/2021” | false |
| 4 | * Number of days in February for non leap year is 28 | * Create instance of Date class with month= 2, day > 28, and year is non leap year * Test data: “2/29/2018” | false |
| 5 | * Number of days in February for leap year is 29 | * Create instance of Date class with month=2, day>29,and year is leap year * Test data: “2/31/2020” | false |
|  |  |  |  |

**Timeslot.compareTo() Test Cases**

| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| --- | --- | --- | --- |
| 1 | * Method must return -1 if one date is less than the other | * Create 2 instances of timeslot where one date comes before the other in terms of year * Test data: “2/14/2021 9:40” and “2/14/2023 8:00 | -1 |
| 2 | * Method must return -1 if one date is less than the other | * Create 2 instances of timeslot where one date comes before the other in terms of month * Test data: “1/14/2021 10:40” and “3/14/2021 7:00 | -1 |
| 3 | * Method must return -1 if one date is less than the other | * Create 2 instances of timeslot where one date comes before the other in terms of day * Test data: “6/10/2020 10:40” and “6/14/2020 7:00 | -1 |
| 4 | * Method must return 1 if one date is greater than the other | * Create 2 instances of timeslot where one date comes after the other in terms of year * Test data: “5/12/2020 10:50” and “5/12/2019 12:00 | 1 |
| 5 | * Method must return 1 if one date is greater than the other | * Create 2 instances of timeslot where one date comes after the other in terms of month * Test data: “4/20/2019 10:50” and “5/20/2019 12:00 | 1 |
| 6 | * Method must return 1 if one date is greater than the other | * Create 2 instances of timeslot where one date comes after the other in terms of day * Test data: “5/20/2019 10:50” and “5/12/2019 12:00 | 1 |
| 7 | * Method must return 0 if both dates and time are equal | * Create 2 instances of timeslot where both date and time are the same * Test data: “1/13/2021 10:50” and “1/13/2021 10:50 | 0 |
| 8 | * Method must return 1 if dates are equal, but time is greater than other time. | * Create 2 instances of timeslot where both dates are equal, and first time is greater than the other in terms of minute * Test data: “12/12/2020 2:10” and “12/12/2020 2:01 | 1 |
| 9 | * Method must return 1 if dates are equal, but time is greater than other time. | * Create 2 instances of timeslot where both dates are equal, and first time is greater than the other in terms of hour * Test data: “12/12/20 3:01” and “12/12/20 2:01 | 1 |
| 10 | * Method must return -1 if dates are equal, but time is less than other time. | * Create 2 instances of timeslot where both dates are equal, and first time is less than the other in terms of hour * Test data: “12/12/20 6:21” and “12/12/20 7:21 | -1 |
| 11 | * Method must return -1 if dates are equal, but time is less than other time. | * Create 2 instances of timeslot where both dates are equal, and first time is less than the other in terms of minute * Test data: “12/12/20 6:21” and “12/12/20 6:40 | -1 |