| Java class/method name being tested: Date class, isValid() method | | | |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test description and Input Data | Expected result |
| 1 | Number of days in February for a non-leap year is 28. | * Create instance of Date with month = 2, day > 28, on a non-leap year * Test data: “2/29/2018” | false |
| 2 | Number of days in February for a leap year is 29. | * Create instance of Date with month = 2, day = 29, on a leap year. * Test data: “2/29/2020” | true |
| 3 | Valid range for the month is 1-12. | * Create instance of Date with the month = 0, day = 1, year = 2022. * Test data: “0/1/2022” | false |
| 4 | Valid range for the month is 1-12. | * Create instance of Date with the month = -1, day = 1, year = 2022. * Test data: “-1/1/2022” | false |
| 5 | Valid range for the month is 1-12. | * Create instance of Date with the month = 13, day = 1, year = 2022. * Test data: “13/1/2022” | false |
| 6 | Valid range for days of the month is no less than 1. | * Create instance of Date with month = 1, day = 0, year = 2022. * Test data: “1/0/2022” | false |
| 7 | Valid range for January is 31 days | * Create instance of Date with month = 1, day = 32, year = 2022. * Test data: “1/32/2022” | false |
| 8 | Valid range for March is 31 days | * Create instance of Date with month = 3, day = 32, year = 2022. * Test data: “3/32/2022” | False |
| 9 | Valid range for May is 31 days | * Create instance of Date with month = 5, day = 32, year = 2022. * Test data: “4/32/2022” | false |
| 10 | Valid range for July is 31 days | * Create instance of Date with month = 7, day = 32, year = 2022. * Test data: “7/32/2022” | false |
| 11 | Valid range for August is 31 days | * Create instance of Date with month = 8, day = 32, year = 2022. * Test data: “8/32/2022” | false |
| 12 | Valid range for October is 31 days | * Create instance of Date with month = 10, day = 32, year = 2022. * Test data: “10/32/2022” | false |
| 13 | Valid range for December is 31 days | * Create instance of Date with month = 12, day = 32, year = 2022. * Test data: “12/32/2022” | false |
| 14 | Valid range for April is 30 days | * Create instance of Date with month = 4, day = 31, year = 2022. * Test data: “4/31/2022” | false |
| 15 | Valid range for June is 30 days | * Create instance of Date with month = 6, day = 31, year = 2022. * Test data: “6/31/2022” | false |
| 16 | Valid range for September is 30 days | * Create instance of Date with month = 9, day = 31, year = 2022. * Test data: “9/31/2022” | false |
| 17 | Valid range for November is 30 days | * Create instance of Date with month = 11, day = 31, year = 2022. * Test data: “11/31/2022” | false |

| Java class/method name being tested: Member class, compareTo() method | | | |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test description and Input Data | Expected result |
| 1 | Compare two members with first and last names equal | * Create instance of Member with finame = “John”,   lname = “Doe”.   * Create instance of Member with fname = “John”,   lname = “Doe”.   * Compare the two members | 0 |
| 2 | Compare two members with first and last names equal | * Create instance of Member with finame = “John”,   lname = “Doe”.   * Create instance of Member with fname = “john”,   lname = “doe”.   * Compare the two members | 0 |
| 3 | Compare two members with first and last names equal | * Create instance of Member with finame = “john”,   lname = “doe”.   * Create instance of Member with fname = “John”,   lname = “Doe”.   * Compare the two members | 0 |
| 4 | Compare two members with m1.fname < m2.fname and m1.lname < m2.lastName | * Create instance of Member with finame = “April”,   lname = “March”.   * Create instance of Member with fname = “Bill”,   lname = “Scanlan”.   * Compare the two members | -6 |
| 5 | Compare two members with m1.fname > m2.fname and m1.lname > m2.lastName | * Create instance of Member with finame = “Bill”,   lname = “Scanlan”.   * Create instance of Member with fname = “April”,   lname = “March”.   * Compare the two members | 6 |
| 6 | Compare two members with m1.fname > m2.fname and m1.lname == m2.lastName | * Create instance of Member with finame = “John”,   lname = “Doe”.   * Create instance of Member with fname = “Jane”,   lname = “Doe”.   * Compare the two members | 14 |
| 7 | Compare two members with m1.fname < m2.fname and m1.lname == m2.lastName | * Create instance of Member with finame = “Jane”,   lname = “Doe”.   * Create instance of Member with fname = “John”,   lname = “Doe”.   * Compare the two members | -14 |
| 8 | Compare two members with m1.fname == m2.fname and m1.lname > m2.lastName | * Create instance of Member with finame = “John”,   lname = “Doe”.   * Create instance of Member with fname = “John”,   lname = “Coe”.   * Compare the two members | 1 |
| 9 | Compare two members with m1.fname == m2.fname and m1.lname < m2.lastName | * Create instance of Member with finame = “John”,   lname = “Coe”.   * Create instance of Member with fname = “John”,   lname = “Doe”.   * Compare the two members | -1 |