|  |  |  |  |
| --- | --- | --- | --- |
| **Java class name being tested: Date class** | | | |
| Java method name being tested: **isValid() method** | | | |
| **Test Case #** | **Requirement** | **Test description and input data** | **Expected result/output** |
| 1 | Any date that has a year <= 0, a month <= 0 or > 12, and a day <= 0 or > 31 is invalid | Create an instance of Date with an invalid year, month, or day.  Test Data:   * “10/1/-1”, * “-2/1/2022” * “13/1/2022” * “10/-1/2022” * “10/32/2022” | **false** |
| 2 | Any short month (30 days) should have a day that is less than 31 | Create an instance of Date with a short month (4, 6, 9, 11), but with a day >= 31  Test Data:   * 11/31/2022 | **false** |
| 3 | Max 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:   * 2/29/2100 | **false** |
| 4 | Max 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:  2/29/2020 | **true** |
| 5 | Any other normal Date should be valid | Create an instance of Date with a valid year, month and day.  Test Data:   * 10/3/2022 | **true** |
| Java method name being tested: **compareTo() method** | | | |
| 1 | Any date that is **before** the current day (the day this class is currently storing) will return **1** | Create an instance of Date with the date: 10/3/2022, and compare it to a **past** date  Test Data:   * “10/2/1997” | **1** |
| 2 | Any date that is **equal** to the current will return **0** | Create an instance of Date with the date: 10/3/2022, and compare it to another instance with the **same** date  Test Data:   * “10/3/2022” | **0** |
| 3 | Any date that is **after** the current day will return **-1** | Create an instance of Date with the date: 10/3/2022, and compare it to a **future** date  Test Data:   * “11/4/2050” | **-1** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Java class name being tested: Member class** | | | |
| Java method name being tested: **compareTo() method** | | | |
| **Test Case #** | **Requirement** | **Test description and input data** | **Expected result/output** |
| 1 | When **Member.CompareMode** is set to **None**, no comparison should be made | Create two instance of Member with random info,  **CompareMode = None**  Test Data (Member):   * Kiana Kaslana, 12/7/1998, 3/30/2717, Bridgewater * Raiden Mei, 4/13/1997, 3/30/2500, Edison | **0** |
| 2 | When **Member.CompareMode** is set to **Name**, any member that the same last name but a first name that is lexicographically greater than the current one will return -1 | Create one instance of Member with a random info, another one but with the same last name and a lexicographically greater first name than the first one,  **CompareMode =** **Name**  Test Data (Member):   * Kiana Kaslana, 12/7/1998, 3/30/2717, Bridgewater * Siegfried Kaslana, 11/25/1972, 7/19/2212, Piscataway | **-1** |
| 3 | When **Member.CompareMode** is set to **Name**, any member that the same last name but a first name that is lexicographically smaller than the current one will return 1 | Create one instance of Member with a random info, another one but with the same last name and a lexicographically smaller first name than the first one.  **CompareMode =** **Name**  Test Data (Member):   * Kiana Kaslana, 12/7/1998, 3/30/2717, Bridgewater * Kallen Kaslana, 11/7/1998, 2/30/2717, Franklin | **1** |
| 4 | When **Member.CompareMode** is set to **County**, any member that the same county but a zip code greater than the current one will return -1 | Create one instance of Member with a random info, another one but with the same county and a zip code greater than the first one  **CompareMode =** **County**  Test Data (Member):   * Kiana Kaslana, 12/7/1998, 3/30/2717, Franklin (county: Somerset, zip code: 08873) * Raiden Mei, 4/13/1997, 3/30/2500, Somerville (county: Somerset, zip code: 08876) | **-1** |
| 5 | When **Member.CompareMode** is set to **County**, any member that the same county but a zip code less than the current one will return 1 | Create one instance of Member with a random info, another one but with the same county and a zip code less than the first one  **CompareMode =** **County**  Test Data (Member):   * Kiana Kaslana, 12/7/1998, 3/30/2717, Franklin (county: Somerset, zip code: 08873) * Raiden Mei, 4/13/1997, 3/30/2500, Bridgewater (county: Somerset, zip code: 08807) | **1** |
| 6 | When **Member.CompareMode** is set to **Expiration Date**, any member that has an expiration date after the current member’s will return -1 | Create one instance of Member with a random info, another one but an expiration date after the first one  **CompareMode =** **ExpirationDate**  Test Data (Member):   * Kiana Kaslana, 12/7/1998, 3/30/2717, Franklin * Raiden Mei, 4/13/1997, 4/1/2727, Somerville | **-1** |
| 7 | When **Member.CompareMode** is set to **Expiration Date**, any member that has an expiration date before the current member’s will return 1 | Create one instance of Member with a random info, another one but an expiration date before the first one  **CompareMode =** **ExpirationDate**  Test Data (Member):   * Kiana Kaslana, 12/7/1998, 3/30/2717, Franklin * Raiden Mei, 4/13/1997, 3/21/2707, Somerville | **1** |
| Java method name being tested: **equals() method** | | | |
| 1 | Any member that has the same first, last name (case insensitive), and date of birth will be consider as equal. Otherwise, not equal. | Create an instance of Member with random info, and another one with the same first, last name, and date of birth.  Test Data:   * Kiana Kaslana, 12/7/1998, 3/30/2717, Franklin * kiana kaslana, 12/7/1998, 3/30/2717, Franklin | **true** |