|  |  |  |  |
| --- | --- | --- | --- |
| Class name: Date  Method signature: public boolean isValid() | | | |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | The method shall return false for any date with the year before 1900. | Create an instance of Date with a valid day and month but an invalid year < 1900.  Test input: 2/2/800 | false |
| 2 | Number of days in Ffebruary for a non-leap year shall be 28.  The method shall return false if the date given has 29 days for a non-leap year. | Create an instance of the Date with the month of February, where day > 29 and the year is non-leap year.  Test input: 2/29/2003 | false |
| 3 | The method shall return false for any month value outside the valid range 1 – 12. | Create an instance of Date with a month > 12 or < 1  Test input: 13/15/2020 | false |
| 4 | The method shall return false for any day value outside the valid range for the month. | Create an instance of Date with a day outside the valid range for the month.  Test input: 4/31/2020 | false |
| 5 | The method shall return true for any valid date in a non-leap year . | Create an instance of Date with a valid day, month, and year.  Test input: 02/29/2024 | true |
| 6 | The method shall return true for dates with day within the maximum days range of a month. | Create an instance of Date with with valid ady for a months.  Test input: 07/31/2021 | true |

|  |  |  |  |
| --- | --- | --- | --- |
| Class name: Profile  Method signature: public boolean commpareTo() | | | |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | When last name and first name are the same, this method should compare date of birth and return correct value. | Compare profiles date of birth when last and first name are the same  Test input:   * profile1 = John Doe (12/13/1989) * profile2 = John Doe (1/13/1985) * profile2.compareTo(profile1) | -1  (1/13/1985  < 12/13/1989) |
| 2 | When last name is the same, then this method should compare first name and return correct value. | Compare profiles by first name when last name is the same.  Test input:   * profile3 = Jane Doe (5/1/1985) * profile2 = John Doe (1/13/1985) * profile3.compareTo(profile2) | -1  (Jane  <  John) |
| 3 | This method should compare last name first and return correct value. | Compare profiles by last name first.  Test input:   * profile3 = Jane Doe (5/1/1985) * profile 4 = Jane March(1/20/2004) * profile3.compareTo(profile4) | -1  (Doe  <  March) |
| 4 | This method should return 0 if compare two identical profiles. | Compare two identical profiles.  Test input:   * profile 5 = Ellington Duke (1/20/2003) * profile 6 = Ellington Duke (1/20/2003) * profile6.compareTo(profile5) | 0 |
| 5 | When last name and first name are the same, this method should compare date of birth and return correct value. | Compare profiles by date of birth when first and last name are the same.  Test input:   * profile1 = John Doe (12/13/1989) * profile2 = John Doe (1/13/1985) * profile1.compareTo(profile2) | 1  (12/13/1989  > 1/13/1985) |
| 6 | Compare profiles by first name when last | Compare profiles by last name first.  Test input:   * profile3 = Jane Doe (5/1/1985) * profile 5 = Ellington Duke (1/20/2003) * profile5.compareTo(profile3) | 1  ( Doe  >  Duke) |
| 7 | When last name is the same, then this method should compare first name and return correct value. | Compare profiles by first name when last name is the same  Test input:   * profile3 = Jane Doe (5/1/1985) * profile2 = John Doe (1/13/1985) * profile2.compareTo(profile3) | 1  ( John  >  Jane ) |