|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Description** | **Sample Input** | **Expected Output** |
| 1 | Test a new date object to check if the date is valid and the toString() to output the date | 12/24/1997 | True  12/24/1997 |
| 2 | Test a leap year date and invoke the isValid() method to show it is a valid date | “02/29/2016” | True |
| 3 | Test the logic of leap year and invokes isValid() method to show it is not valid | “02/29/2017” | False |
| 4 | Test the equals(obj) method by comparing two objects different objects | D1= “12/24/1997”  Feb = “02/29/2016”  D1.equals(feb); | False |
| 5 | Test that letters is not a valid date | “ab/28/2020” | False |
| 6 | Test that invalid punctuation marks are invalid and calls isValid() | “09-11-2021” | False |
| 7 | Invoke the equals() method to compare two objects with the same field entries | Test7 = “12/24/1997”  D1 = “12/24/1997”  Test7.equals(d1); | True |

**Test Design Document**

**Date Test Case** –

**TeamMember Test Case** –

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Description** | **Sample Input** | **Expected Output** |
| 1 | Invoke the TeamMember constructor and toString() to display | “John”, “12/24/1997” | “John 12/24/1997” |
| 2 | Invoke the equals() method to compare two objects with the same name entries but different dates | “John” , “12/23/1997”  “John” , “12/24/1997” | False |
| 3 | Calls the getStartDate() method | “Johnny”, “02/17/2020” | “02/17/2020” |
| 4 | Compares two different objects with the .equals() method | “John”, “12/23/1997”  “Johnny”, “02/17/2020” | False |