| **Method** | **Test Case** | | **Input** | **Expected Output** | **Actal Output** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| Year() | defaultConstructor\_SHOULDNOT\_returnNull | | none | non-null Year object | non-null Year object | ✅ Passed |
| Year() | testYearDefaultCtor | | none | Year with current year | Year with current year | ✅ Passed |
| Year(int) | intParameterConstructor\_SHOILD\_return\_sameGivenArg\_WHEN\_weCallGetYearMethod | | 2025 | 2025 | 2025 | ✅ Passed |
| Year(int) | intParamConstructor\_SHOULD\_throwIllegalArgumentException\_WHEN\_itReceivesBeforeMin | | Below MINIMUM\_YEAR | Exception | Exception | ✅ Passed |
| Year(int) | intParamConstructor\_SHOULD\_throwIllegalArgumentException\_WHEN\_itReceivesBelow1900 | | 1899 | Exception (based on modified doc) | No exception | ❌ Failed |
| Year(Date) | dateParamConstructor\_SHOULD\_return\_sameGivenArg\_WHEN\_weCallGetDateMethod | | new Date(2025-1900, 0, 1) | 2025 | 2025 | ✅ Passed |
| Year(Date) | dateParamConstructor\_SHOULD\_throwNullPointerException\_WHEN\_receivesNullDate | | null | NullPointerException | Exception | ✅ Passed |
| Year(Date, TZ, L) | Year\_SHOULD\_extractYearCorrectly\_WHEN\_createdWithValidDateAndDefaults | | 2025-1900, 0, 1 + defaults | 2025 | 2025 | ✅ Passed |
| compareTo() | compareTo\_SHOULD\_returnNegative | | 2024.compareTo(2025) | -1 | -1 | ✅ Passed |
| compareTo() | compareTo\_SHOULD\_returnPositive | | 2026.compareTo(2025) | 1 | 1 | ✅ Passed |
| compareTo() | compareTo\_SHOULD\_throwNullPointerException\_WHEN\_inputIsNull | | null | NullPointerException | Exception | ✅ Passed |
| equals() | equals\_SHOULD\_returnTrue\_WHEN\_receiveSameYear | | 2025.equals(2025) | true | true | ✅ Passed |
| equals() | equals\_SHOULD\_returnFalse\_WHEN\_receiveDifferentYear | | 2025.equals(2026) | false | false | ✅ Passed |
| getFirstMillisecond | getFirstMillisecond\_SHOULD\_returnFirstMillisecond\_WHEN\_receiveSameCalenderYear | | 2025 + Calendar | correct start ms | correct | ✅ Passed |
| getFirstMillisecond | getFirstMillisecond\_SHOULD\_throwNullPointerException\_WHEN\_receivesNoCalenderYear | | null | NullPointerException | Exception | ✅ Passed |
| getLastMillisecond | getLastMillisecond\_SHOULD\_returnLastMillisecond\_WHEN\_receiveSameCalenderYear | | 2025 + Calendar | correct end ms | correct | ✅ Passed |
| getSerialIndex() | getSerialIndex\_SHOULD\_returnIndex\_WHEN\_receiveSameYearValue | 2025 | | 2025 | 2025 | ✅ Passed |
| hashCode() | hashCode\_SHOULD\_returnSameHash\_WHEN\_receiveSameYearValue | | two Year(2025) | same hash | same | ✅ Passed |
| hashCode() | hashCode\_SHOULD\_returnDifferentHash\_WHEN\_receiveDifferentYearValue | | 2025, 2026 | different hash | different | ✅ Passed |
| next() | next\_SHOULD\_returnNextYear\_WHEN\_receivesAnyYearValue | | 2025 | Year(2026) | Year(2026) | ✅ Passed |
| next() | next\_SHOULD\_returnNull\_WHEN\_yearIsMaximumAllowed | | 9999 | null | null | ✅ Passed |
| parseYear() | parseYear\_SHOULD\_returnYearObject\_WHEN\_receivesYearValue | | "2025" | Year(2025) | Year(2025) | ✅ Passed |
| parseYear() | parseYear\_SHOULD\_returnNull\_WHEN\_inputIsNonNumeric | | "abcd" | null | null | ❌ Faild |
| parseYear() | parseYear\_SHOULD\_returnNull\_WHEN\_inputIsEmptyString | | "" | null | null | ❌ Fail) |
| parseYear() | parseYear\_SHOULD\_throwNullPointerException\_WHEN\_receivesNullYearValue | | null | NullPointerException | Exception | ✅ Passed |
| peg() | peg\_SHOULD\_shouldEvaluateToStartAndEndTime\_WHEN\_receivesValidStartAndEndCalenderYear | | Calendar | correct start/end dates | correct | ✅ Passed |
| previous() | previous\_SHOULD\_returnPreviousYear\_WHEN\_receivesAnyYearValue | | 2025 | Year(2024) | Year(2024) | ✅ Passed |
| previous() | previous\_SHOULD\_returnNegativePrevious\_WHEN\_receiveZero | | 0 → previous() | Year(-1) | Year(-1) | ✅ Passed |
| previous() | previous\_SHOULD\_returnNull\_WHEN\_reeivesPrev1900 | | 1900 → previous() | null (based on modified doc) | not null (Year 1899) | ❌ Failed |
| toString() | toString\_SHOULD\_returnString\_WHEN\_receivesYearValue | | -500 | "-500" | "-500" | ✅ Passed |
| toString() | toString\_SHOULD\_throwNullPointerException\_WHEN\_calledOnNull | | null reference | NullPointerException | Exception | ✅ Passed |

Bug Report for **Discount manager** Methods

**Method: calculatePriceAfterDiscount(double price)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Input** | **Expected Output** | **Actual Output** | **Status** | **Comment** |
| 1 | isDiscountSeason=false ,price=100 | 100 | 100 | passed | No Comment |
| 2 | isDiscountSeason=true, specialWeek=true ,price=200 | 160 | 160 | passed | No Comment |
| 3 | isDiscountSeason=true, specialWeek=false ,discount = 9 ,price=150 | 13.5 | 1350.0 | Fail | Wrong calculation in getDiscountPercentage |

Bug Report for **Discount calculator** Methods

**Method: isTheSpecialWeek(), getDiscountPercentage()**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Input** | **Expected Output** | **Actual Output** | **Status** | **Method** |
| 1 | Feb 22, 2025 | false | false | passed | isTheSpecialWeek() |
| 2 | June 24, 2025 | True | True | passed | isTheSpecialWeek() |
| 3 | Jan 7, 2025 | 7 | 7 | passed | getDiscountPercentage() |
| 4 | Jan 1, 2025 | 5 | 5 | passed | getDiscountPercentage() |