

IT-314

Lab10: – Specification-based Test Case Generation

Q1: Consider a program for determining the previous date. Its input is triple of day, month and year with the following ranges $1 \leq \text{month} \leq 12$, $1 \leq \text{day} \leq 31$, $1900 \leq \text{year} \leq 2015$. The possible output dates would be previous date or invalid date. Design the equivalence class test cases?

Write a set of test cases (i.e., test suite) – specific set of data – to properly test the programs. Your test suite should include both correct and incorrect inputs.

1. Enlist which set of test cases have been identified using Equivalence Partitioning and Boundary Value Analysis separately.
2. Modify your programs such that it runs on eclipse IDE, and then execute your test suites on the program. While executing your input data in a program, check whether the identified expected outcome (mentioned by you) is correct or not.

Equivalence classes:

Month:

M1: Months with 31 days(Jan(01),March(03),May(05),July(07),Aug(08),Oct(10),
Dec(12))

M2: Months with 30 days(April(04),June(06),Sept(09),November(11))

M3: Month with 28/29 days(Feb(02))

M4: Month number<1

M5: Month number>12

Days

D1:1 to 28

D2: 29

D3: 30

D4: 31

D5: <1

D6: >31

Years

Y1: Leap years from 1900 to 2015

Y2: Non-Leap years from 1900 to 2015

Y3: >2015

Y4: <1900

ID	Days	Months	Year	Output
1	D1	M1	Y1	Previous Date
2	D2	M1	Y1	Previous Date
3	D3	M1	Y1	Previous Date
4	D4	M1	Y1	Previous Date
5	D5	Any	Any	Invalid date
6	D6	Any	Any	Invalid date
7	D1	M1	Y2	Previous date
8	D2	M1	Y2	Previous date
9	D3	M1	Y2	Previous date
10	D4	M1	Y2	Previous date
11	ANY	ANY	Y3	Invalid
12	ANY	ANY	Y4	Invalid
13	D1	M2	Y1	Previous Date
14	D2	M2	Y1	Previous Date
15	D3	M2	Y1	Previous Date
16	D4	M2	Y1	Invalid
17	D1	M2	Y2	Previous Date
18	D2	M2	Y2	Previous Date
19	D3	M2	Y2	Previous Date
20	D4	M2	Y2	Invalid
21	D1	M3	Y1	Previous Date
22	D2	M3	Y1	Previous Date
23	D3	M3	ANY	Invalid
24	D4	M3	ANY	Invalid
25	D1	M3	Y2	Previous Date

26	D2	M3	Y2	Previous Date
27	ANY	M4	ANY	Invalid
28	ANY	M5	ANY	Invalid

Test case table:

Test date	ID of equivalence class	Expected output	Methodology
12-03-2000	1	Previous date	Equivalence Partitioning
29-03-2000	2	Previous date	Equivalence Partitioning
30-03-2000	3	Previous date	Boundary Value Analysis
31-03-2004	4	Previous date	Boundary Value Analysis
0-01-2005	5	Invalid date	Boundary Value Analysis
33-05-2013	6	Invalid date	Equivalence Partitioning
25-11-2005	7	Previous date	Equivalence Partitioning
29-01-2007	8	Previous date	Equivalence Partitioning
30-07-1995	9	Previous date	Equivalence Partitioning
31-01-1999	10	Previous date	Boundary Value Analysis
10-04-2021	11	Invalid date	Equivalence Partitioning
10-04-1985	12	Previous date	Equivalence Partitioning
30-11-2004	15	Previous date	Boundary Value Analysis
29-04-1994	18	Previous date	Boundary Value Analysis
31-04-2001	20	Invalid date	Boundary Value Analysis
29-02-2004	26	Previous date	Boundary Value Analysis
01-00-2014	27	Invalid date	Boundary Value Analysis
12-500000-2003	28	Invalid date	Equivalence Partitioning

30-02-1993	23	Invalid date	Boundary Value Analysis
------------	----	--------------	-------------------------

Q2: You are testing an e-commerce system that sells products like caps and jackets. The problem is to create functional tests using boundary-value analysis and equivalence class partitioning techniques for the web page that accepts the orders. A screen prototype for the order-entry web page is shown below.

The system accepts a five-digit numeric item ID number from 00000 to 99999. The system accepts a quantity to be ordered, from 1 to 99. If the user enters a previously ordered item ID and a 0 quantity to be ordered, that item is removed from the shopping cart. Based on these inputs, the system retrieves the item price, calculates the item total (quantity times item price), and adds the item total to the cart total. Due to limits on credit card orders that can be processed, the maximum cart total is \$999.99

Equivalence Classes:

ID:

E1: no<00000

E2: no>99999

E3: Not from(00000-99999), both numbers included.

Qty:

Q1: qty<0

Q2: qty>99

Q3: 0<qty<=99

Q4: qty=0

Cart total:

C1: <=999.99

C2: >999.99

Let's assume a product with ID 11111 has a price of \$250

ID	Qty	Cart Total	Output
E1	Any	Any	Error
E2	Any	Any	Error
Any	Q1	Any	Error
E3	Q4	Any	Error(if the item was not purchased previously)

E3	Q4	C1	Price without the ID item displayed
Any	Q2	Any	Error
E3	Q3	C1	Price displayed
Any	Any	C2	Error