Writing Test Cases II

Cai, Zelin Silvestre, Patrick

1 No Currency Test Cases - Original Test Cases

Test Name	Input Vector	Expected Output
test dispense red	`<dispense_red()>`	“You need at least 5 cents to dispense a red gumball”
test dispense yellow	`<dispense_yellow()>`	“You need at least 10 cents to dispense a red gumball”
test return my change	`<return_my_change()>`	“There is no change to return”

Table 1: Original ”No Currency” Test Cases

Of these three test cases, no particular technique (e.g. control flow testing, data flow testing, etc.) was used in their creation.

2 No Currency Test Cases - Revised Test Cases

In general, the input domain associated with “no currency” testing involve invocations of any functions other than `insert()`.

2.1 Equivalence Class Testing

The input domain of “all functions other than `insert()`” can be partitioned based on the remaining functions:

- EC-01: `return_invalid_currency()`
- EC-02: `dispense_red()`
- EC-03: `dispense_yellow()`
- EC-04: `return_my_change()`

Tests are generated by creating a test case for each uncovered equivalence class:

TC	Test Value	Expected Result	EC-01	EC-02	EC-03	EC-04
01	`<return_invalid_currency()>`	“”	✓			
02	`<dispense_red()>`	“You need at least 5 cents to dispense a red gumball”		✓		
03	`<dispense_yellow()>`	“You need at least 10 cents to dispense a yellow gumball”			✓	
04	`<return_my_change()>`	“There is no change to return”				✓

Table 2: Revised ”No Currency” Test Cases - Equivalence Class Testing

2.2 Boundary Value Analysis