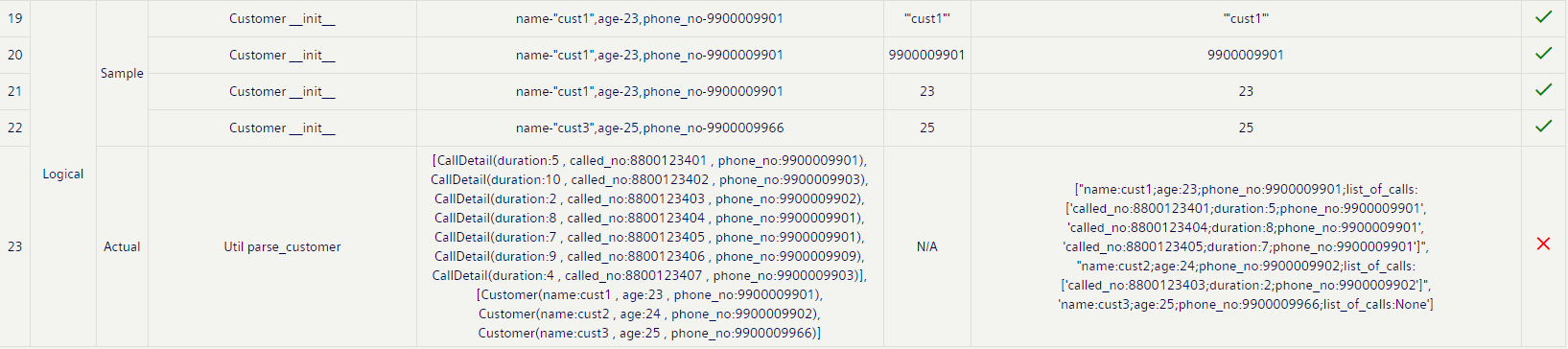
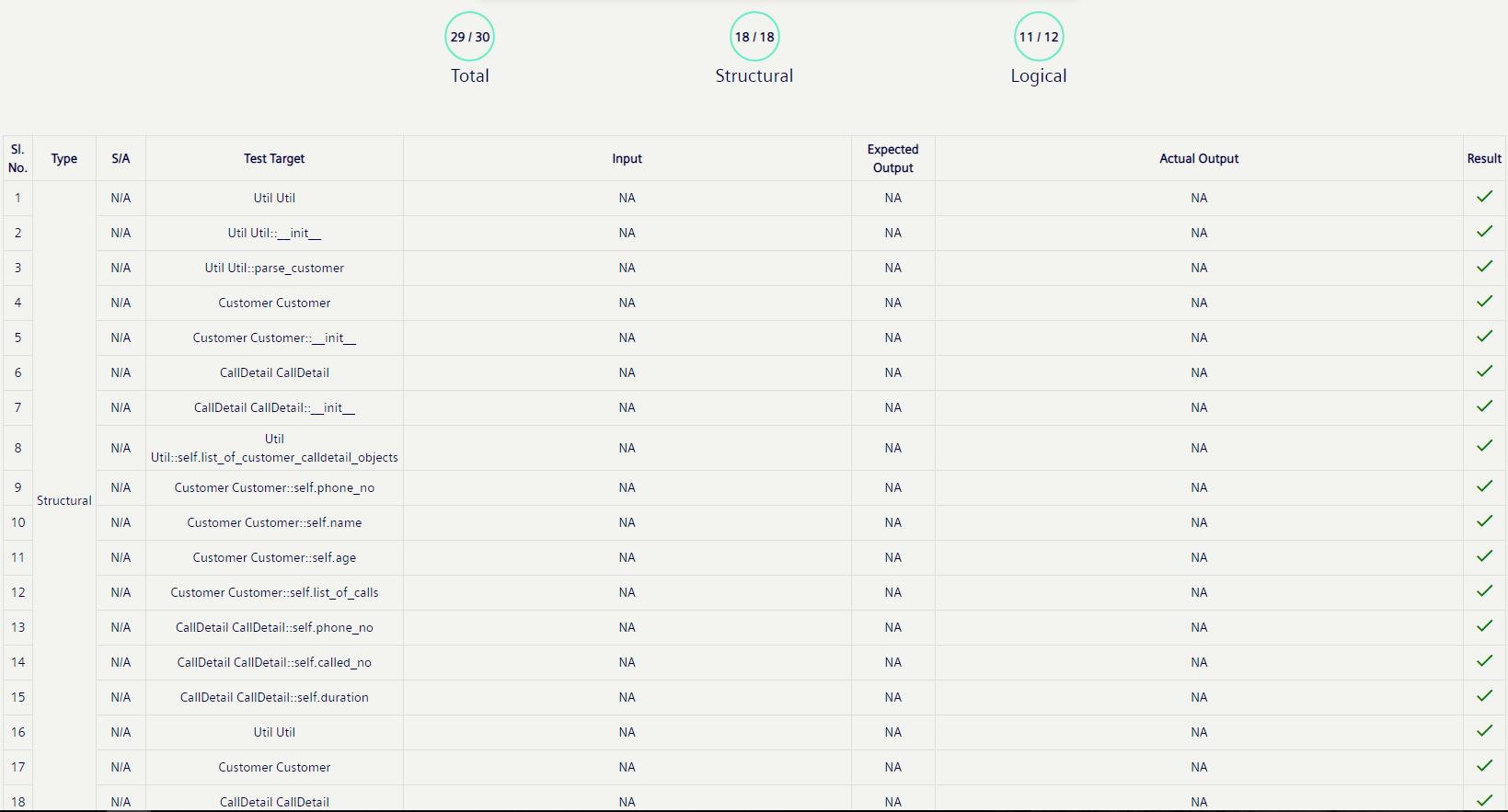
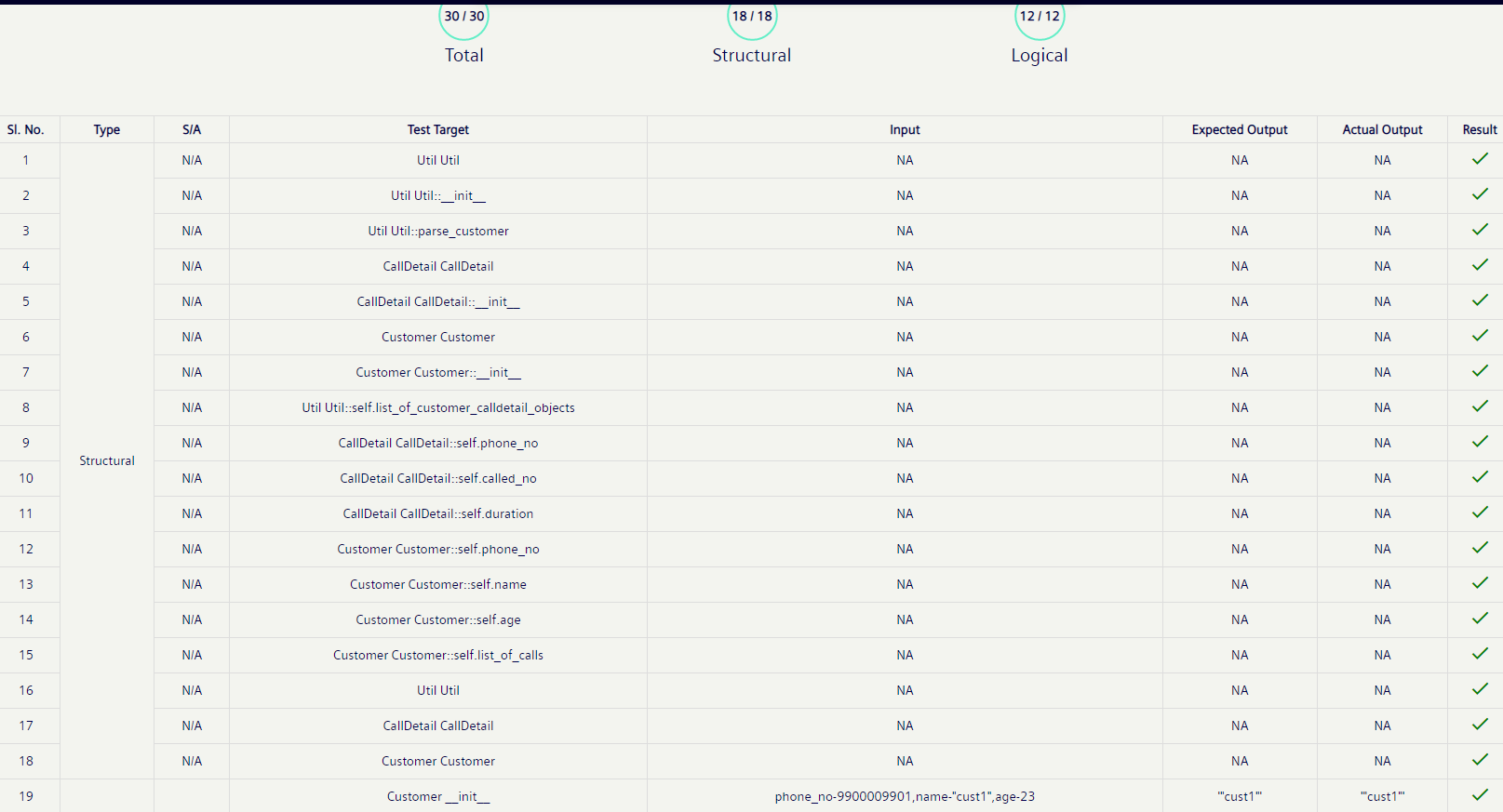
## Assignment on Dependency - Level 2

A telecom company wants to generate reports on the call details of the customers. Each customer can make multiple phone calls.  
  
**Problem Statement:**The parse\_customer method takes a list of Customer objects and a list of CallDetail objects. For every customer, identify all the corresponding Call Detail objects ( the customer phone number and the phone number of Call detail object have to match ), add them to the list\_of\_calls of corresponding customer object and add the updated customer object to list\_of\_customer\_calldetail\_objects of Util class.

1Fail: \**was failing due to my ERROR CHECK - if(len(temp)!=0:*

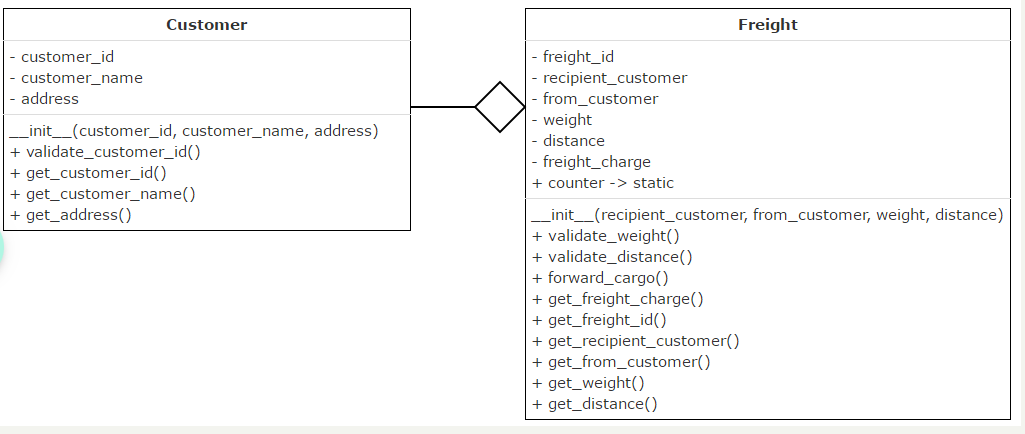


All Pass:



## Assignment on Aggregation - Level 2

Freight Pvt. Ltd, a cargo company, forwards cargos/freights between its customers.  
Freight charges are applied based on weight and distance of the shipment.  
Write a python program to implement the class diagram given below.



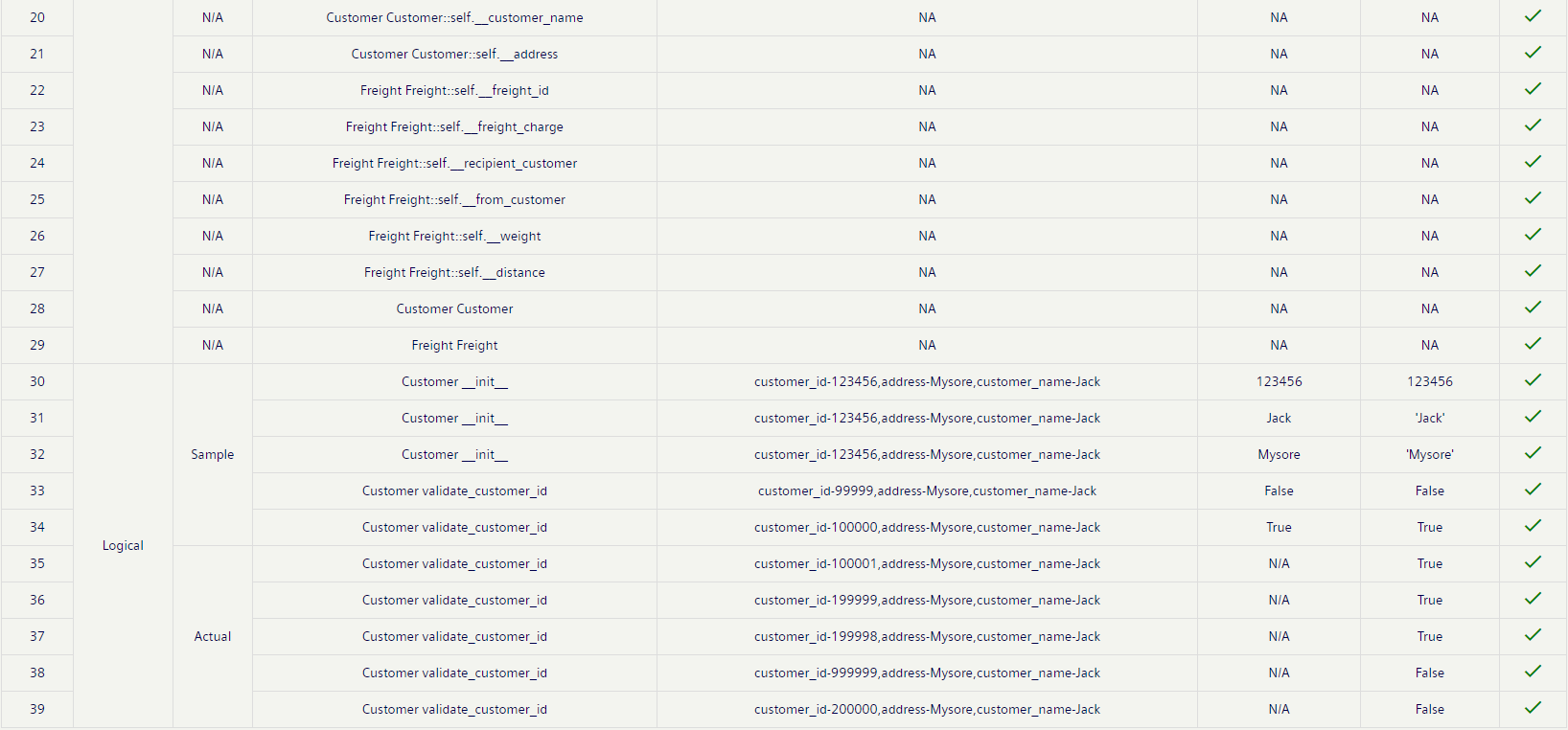
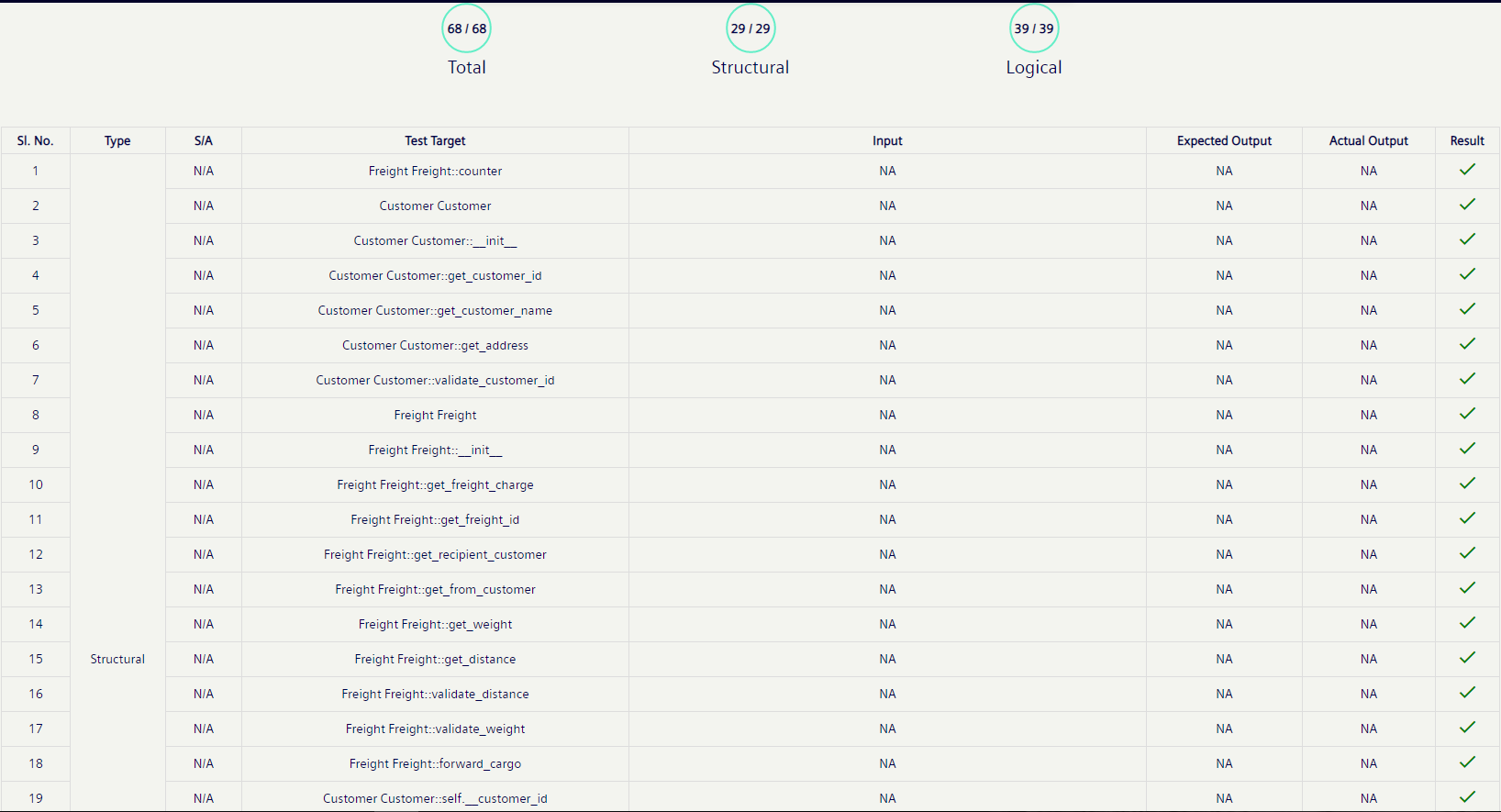
**Method description:**

1. Initialize counter variable to 198 in Freight class
2. All validate methods should return true, if validation succeeds. Else it should return false
3. **validate\_customer\_id():** Customer id should be 6 digits and should begin with digit 1
4. **validate\_weight():** Weight should be a multiple of 5
5. **validate\_distance():**Distance should be between 500kms and 5000kms (both inclusive)
6. **forward\_cargo():**
   * Validate from\_customer.customer\_id, recipient\_customer.customer\_id, distance and weight of the freight
   * If valid,
     + auto-generate freight\_id starting from 200 and initialize it. freight\_id should be even
     + calculate freight\_charge based on weight (Rs.150/kg) and distance (Rs.60/km)
   * Else, set freight\_charge to 0

**For testing:**

* Create objects of Customer and Freight class
* Invoke forward\_cargo() method on Freight object
* Display freight id and freight charge
* In case of error/invalid data, display appropriate error message

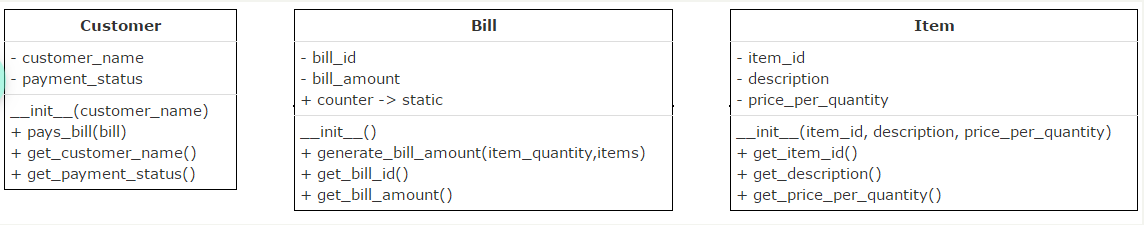
All Pass:



## Assignment on Dependency & List Objects - Level 2

In the retail store scenario, let's look at the portion of customer purchasing items from the retail store.

Write a python program to implement the class diagram given below.



**Class Description:   
Bill class:**

1. Initialize static variable counter to 1000
2. **generate\_bill\_amount(item\_quantity,items):**Calculate bill amount based on the items purchased by the customer
   1. Accept a dictionary, item\_quantity which contains the item id (key) of the items purchased along with the quantity (value)
   2. Accept a list, items which contains the list of Item objects available in the store
   3. Generate bill id starting from 1001 prefixed by "B" and initialize attribute, bill\_id. Ex. "B1001", "B1002" etc.
   4. Calculate bill amount based on the quantity and price of the items purchased by the customer
   5. Set attribute, bill\_amount with the calculated bill amount

Assume that values in item\_quantity and items are always valid.   
**Customer class:**

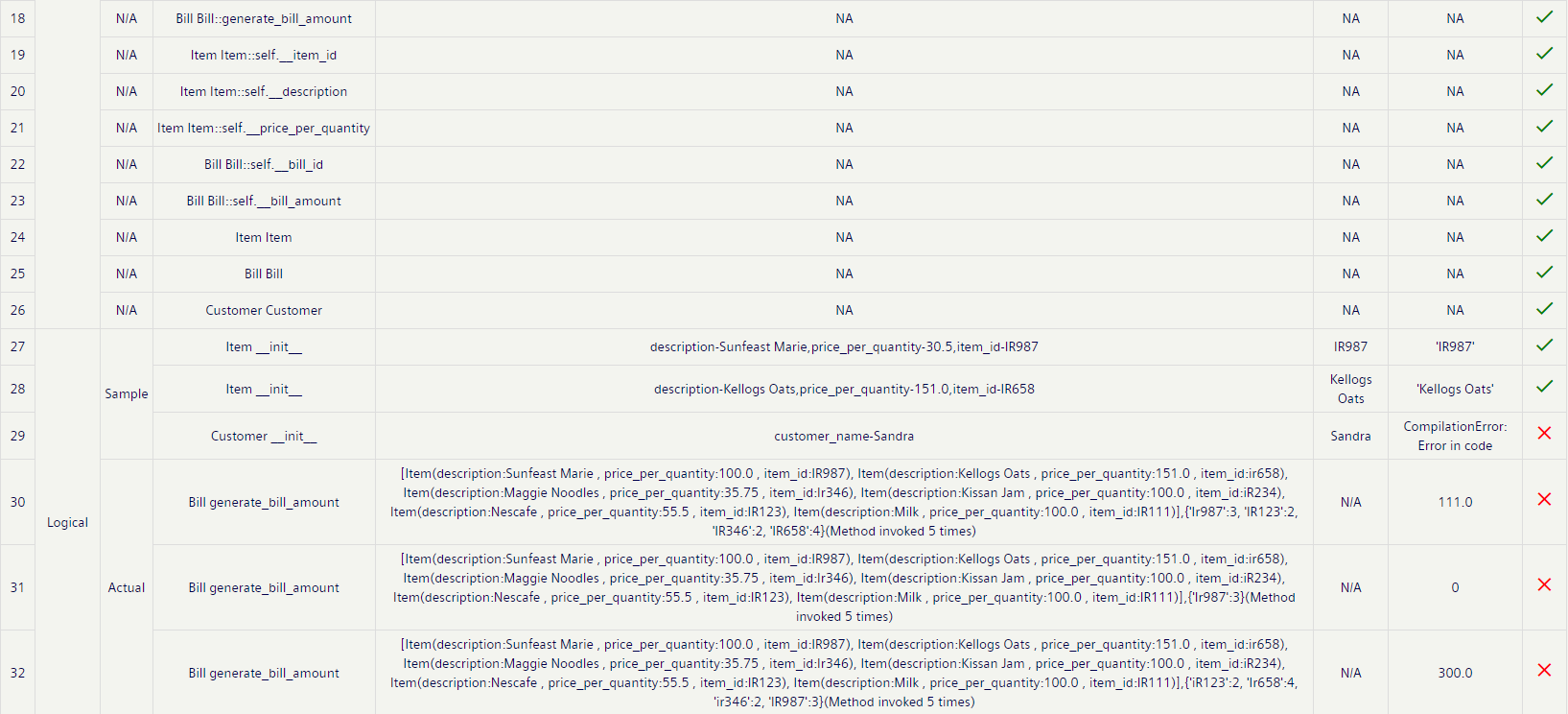
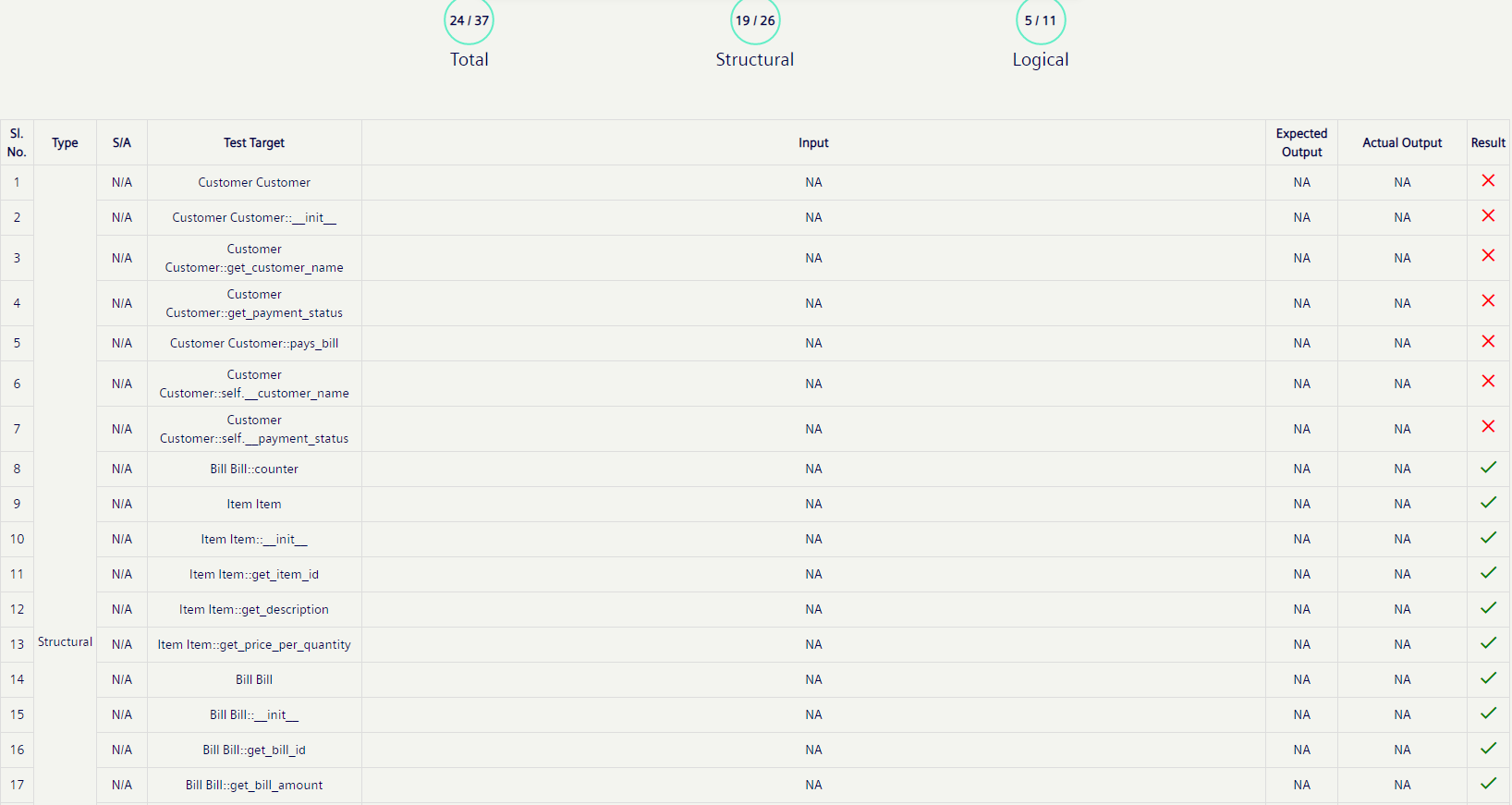
**pays\_bill(bill):**Pay bill based on the bill amount

1. Accept Bill object which contains the details of the bill to be paid by the customer
2. Update attribute, payment\_status to "Paid"
3. Display customer name, bill id and bill amount

**Note:** Perform case insensitive string comparison  
  
**For testing:**

* Create objects of Customer class, Item class and Bill class
* Invoke generate\_bill\_amount(item\_quantity,items) on Bill object by passing the dictionary containing item\_id and quantity of items purchased by the Customer and list of Item objects
* Invoke pays\_bill() on Customer object by passing the Bill object

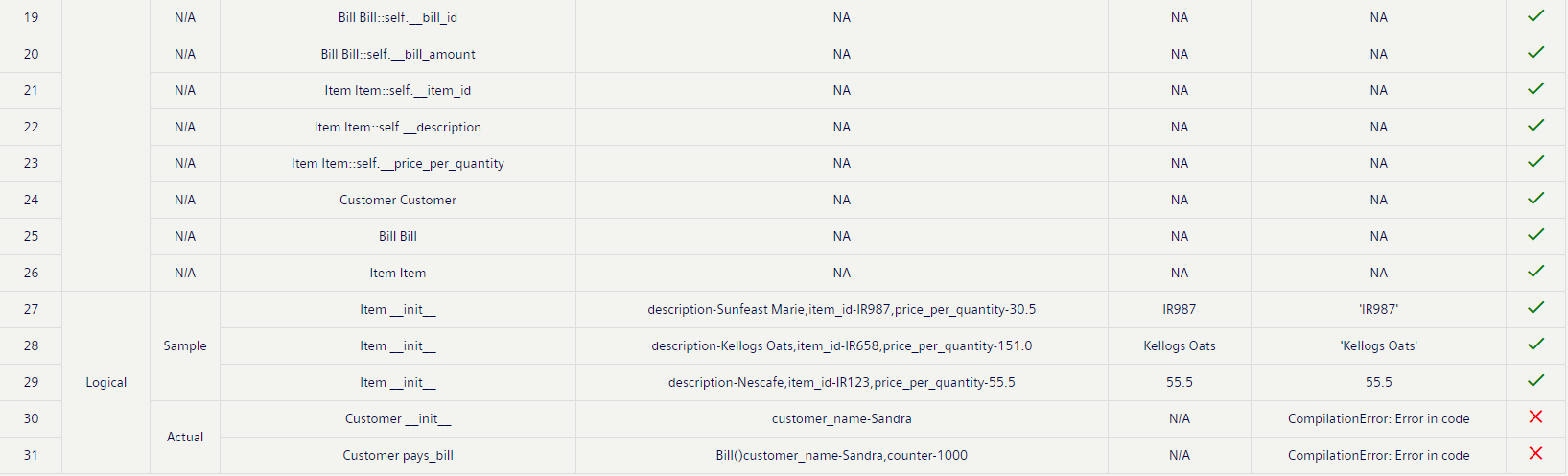
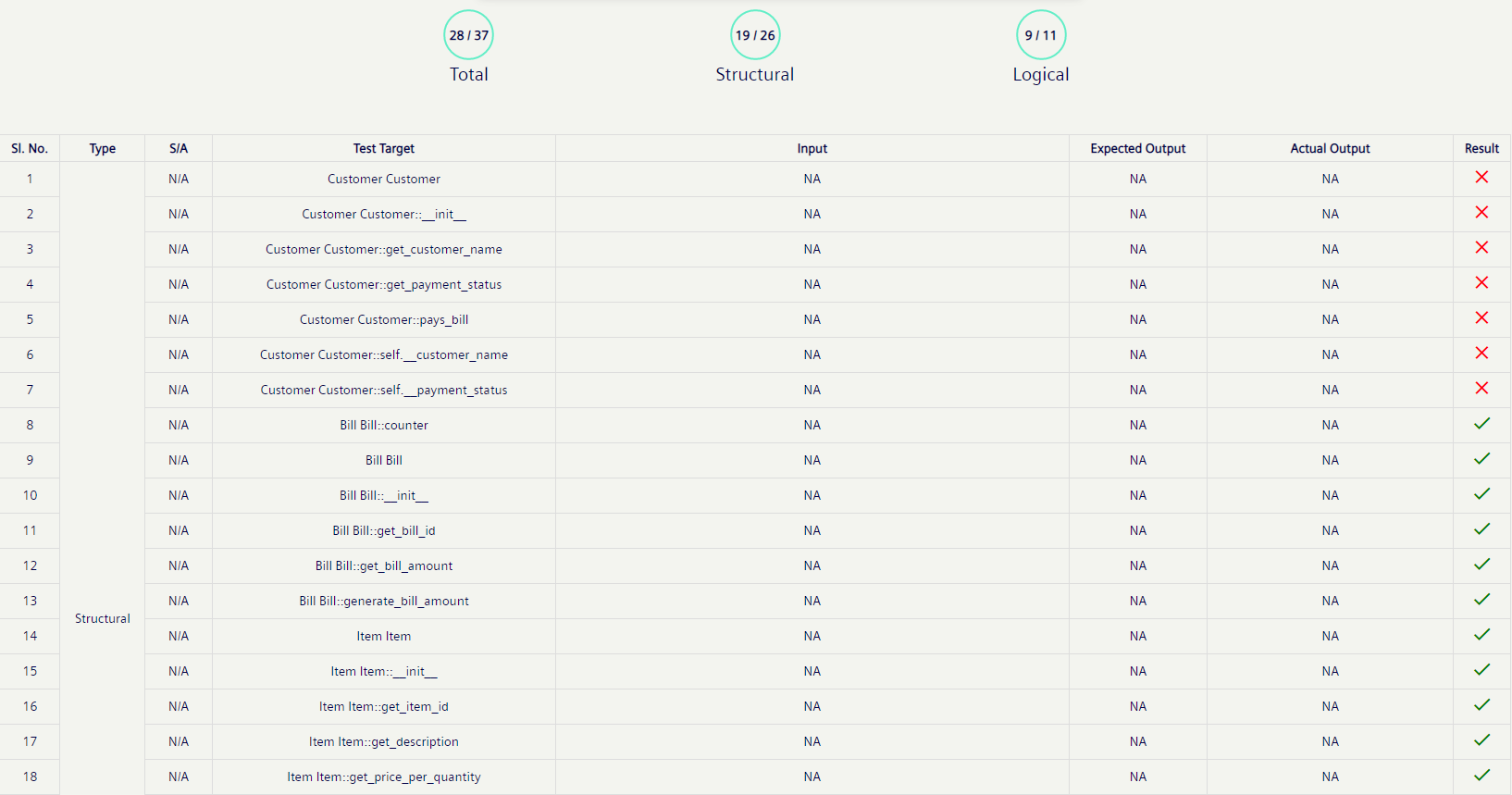
Initial Verification:



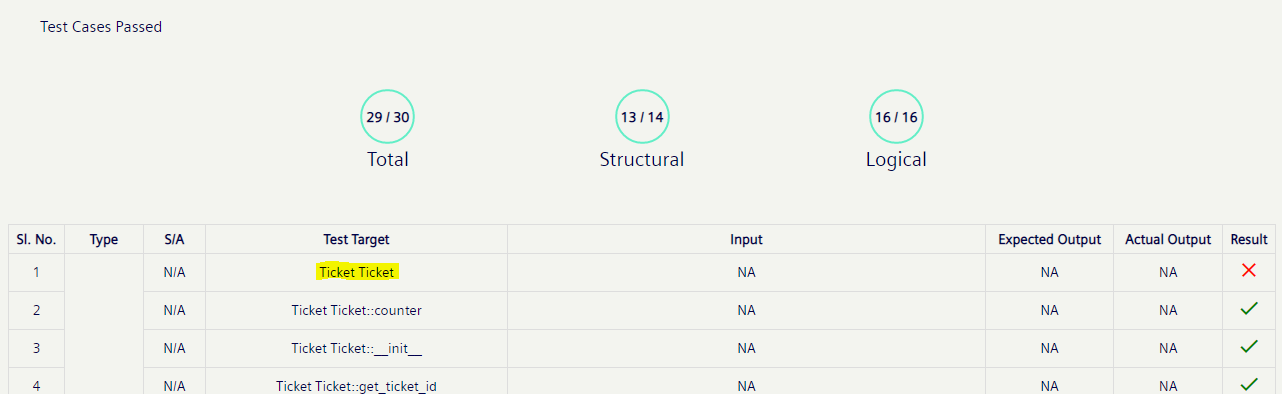
Debugging:

* Bill generate\_bill\_amount test case passed after adding “if(key.upper()==item.get\_item\_id().upper()):” since ***case insensitive*** string comparison

Still Below test cases failing:



## Assignment on Static Counter - Level 2 [Repeated]

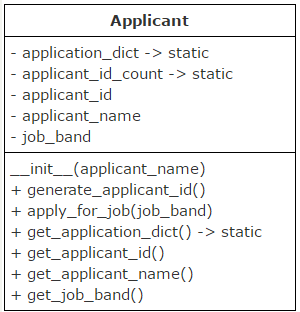


*What is the above Testcase checking for?*

## Assignment on Static - Level 2

"Infinity" IT solution wants to automate their recruitment process. They have decided to accept 5 applications for each of the three job bands ("A", "B" and "C") in the company.

Write a Python program to implement the class diagram given below.



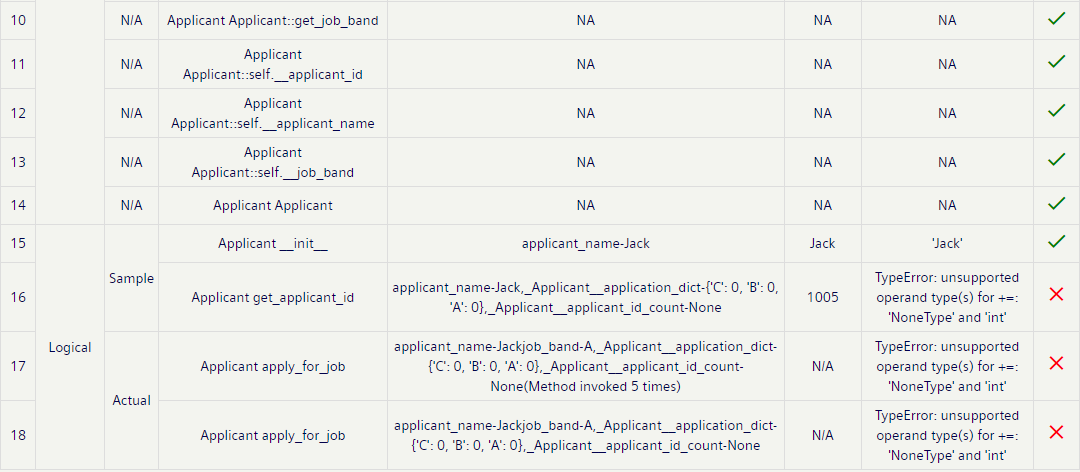
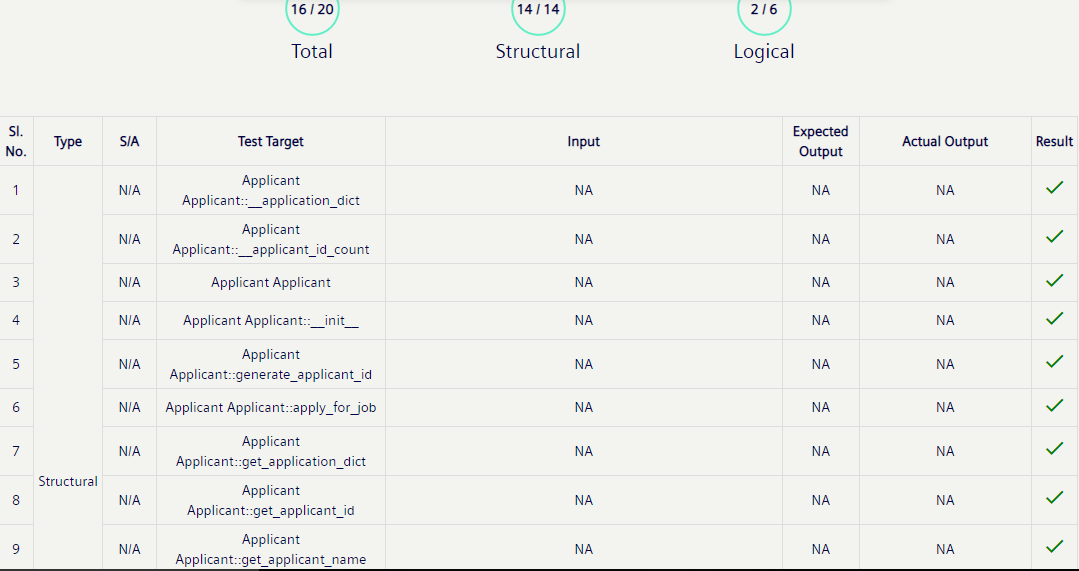
**Method/Attribute description:**

1. Initialize static variable, applicant\_id\_count to 1000
2. **application\_dict:** Dictionary which store application count (value) for each job band (key)
3. **generate\_applicant\_id():** Auto-generate applicant id starting from 1001 and initialize attribute, applicant\_id
4. **apply\_for\_job(job\_band):** Accept the job band for which the applicant is applying.
   1. Check if application count for the applied job band has reached the maximum limit, 5. If so, return -1.
   2. Else,
      * increment application count for the applied job band by 1 in the dictionary
      * generate applicant id and
      * initialize attribute, job\_band with the applied job\_band

**For testing:**

* Create objects of Applicant class
* Invoke apply\_for\_job(job\_band) method on Applicant object by passing the job band for which applicant is applying
* If application is accepted, display applicant id, name and job band
* Else, display appropriate error message

Verification:



\*\*\* All test cases were failing earlier due to the print statement in else: of the method – “def **apply\_for\_job**(*self*,job\_band):”

Still some test cases ; check later.