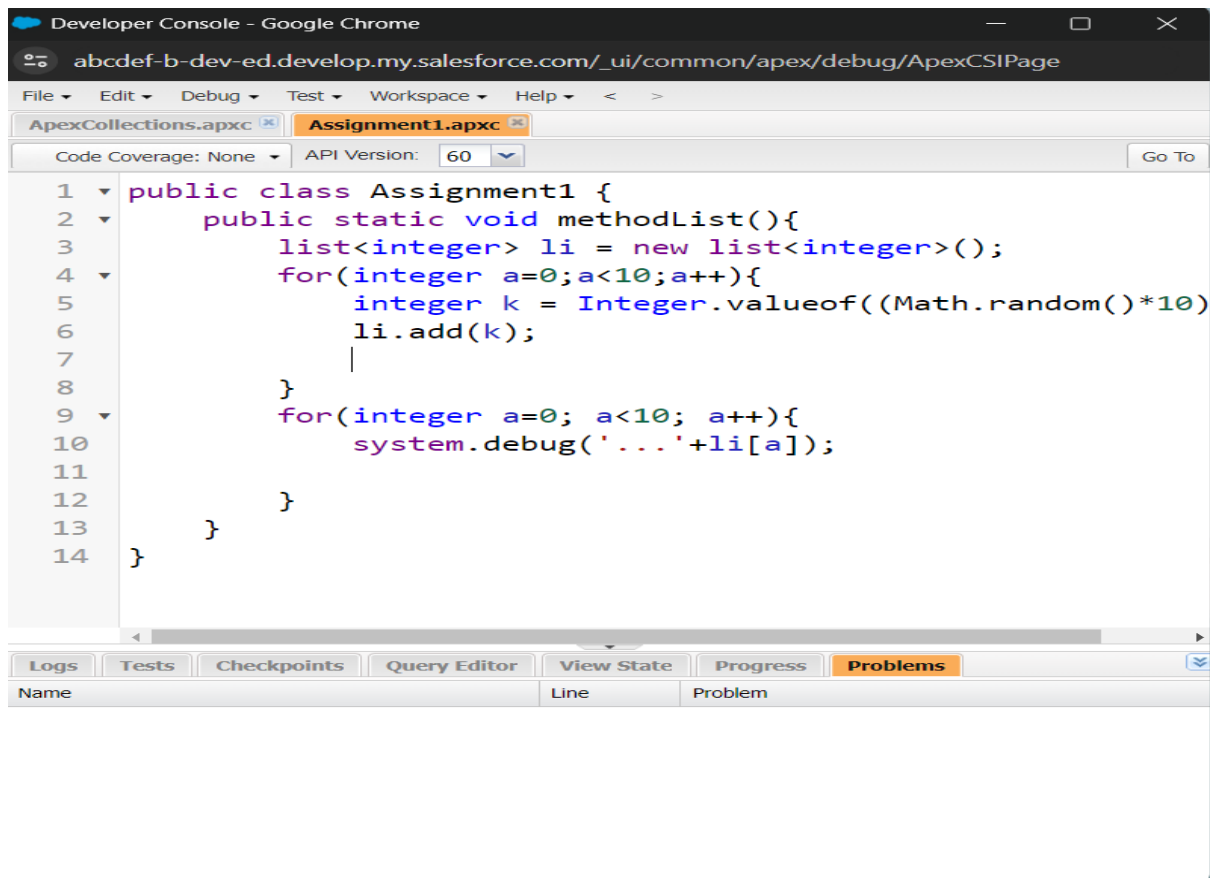


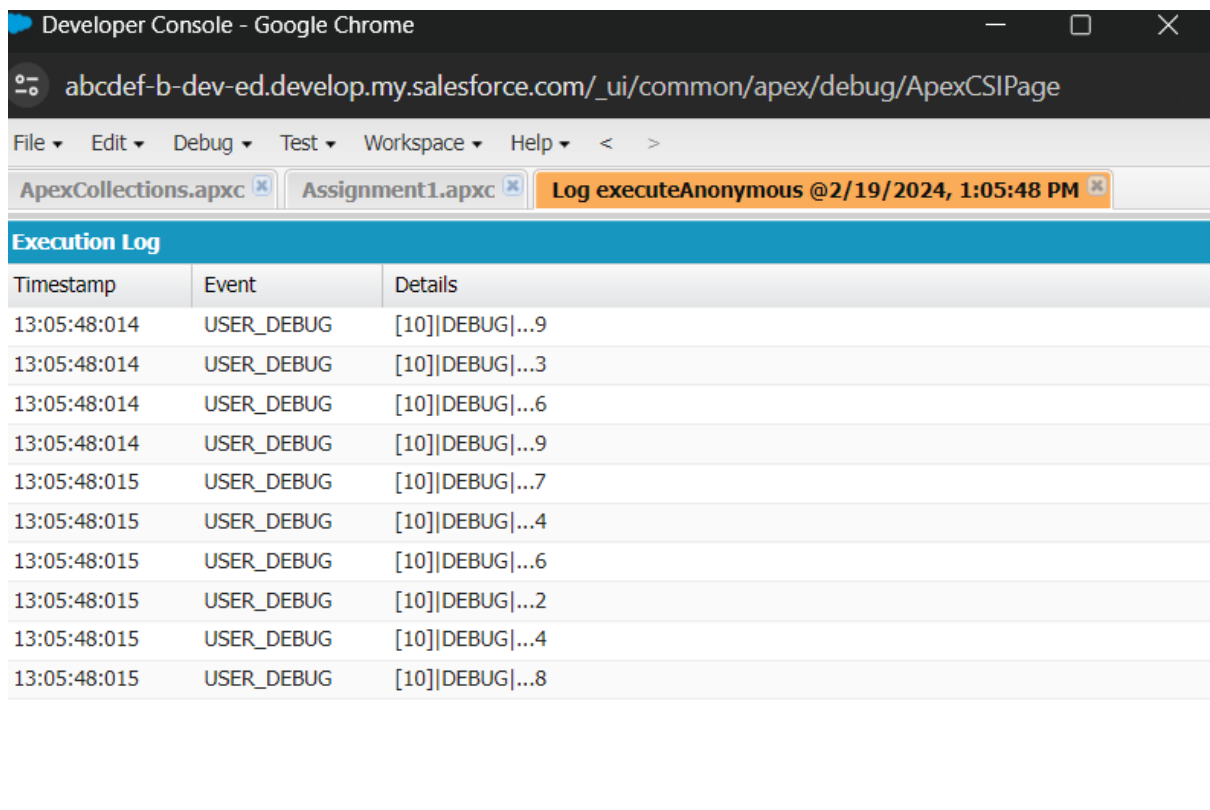
## Collection and SOQL Assignment

- Add the values to the list



The screenshot shows the Salesforce Developer Console with the Apex code for Assignment1.apxc. The code defines a public class Assignment1 with a static method methodList(). Inside methodList(), a list of integers is created and populated with random values. The code is as follows:

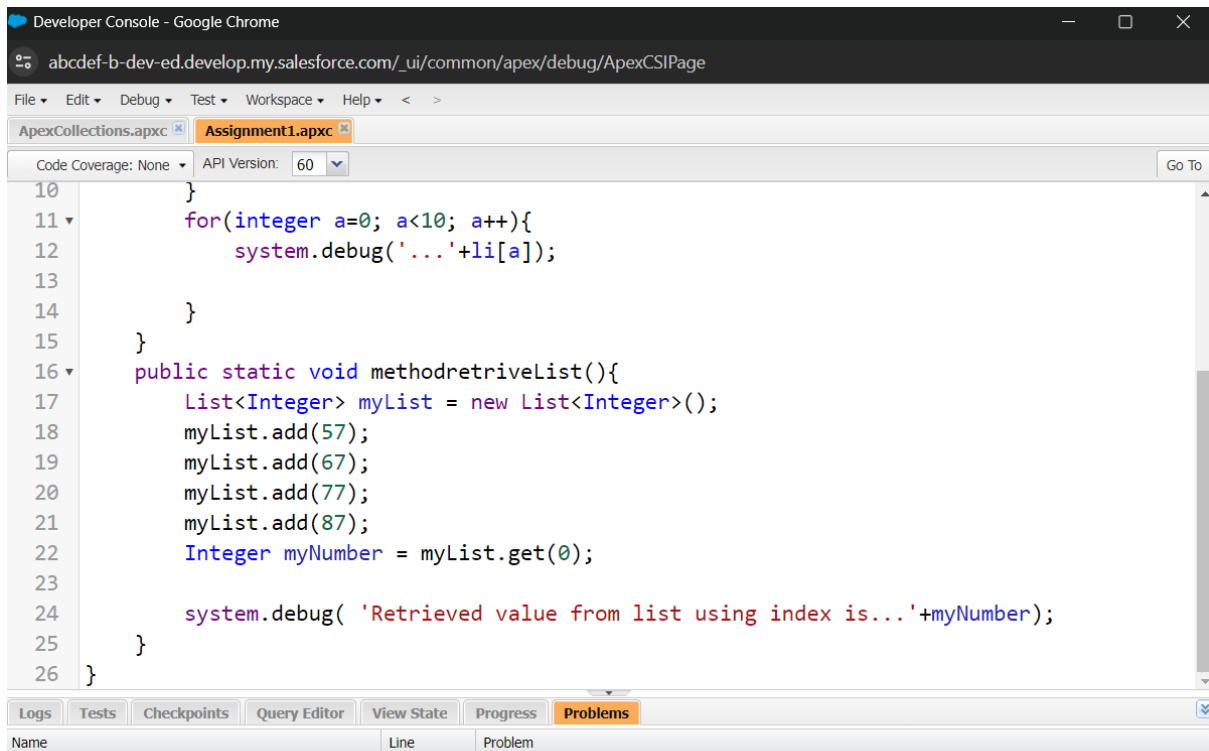
```
1 public class Assignment1 {
2     public static void methodList(){
3         list<integer> li = new list<integer>();
4         for(integer a=0;a<10;a++){
5             integer k = Integer.valueOf((Math.random()*10)
6             li.add(k);
7         }
8     }
9     for(integer a=0; a<10; a++){
10        system.debug('...'+li[a]);
11    }
12 }
13 }
14 }
```



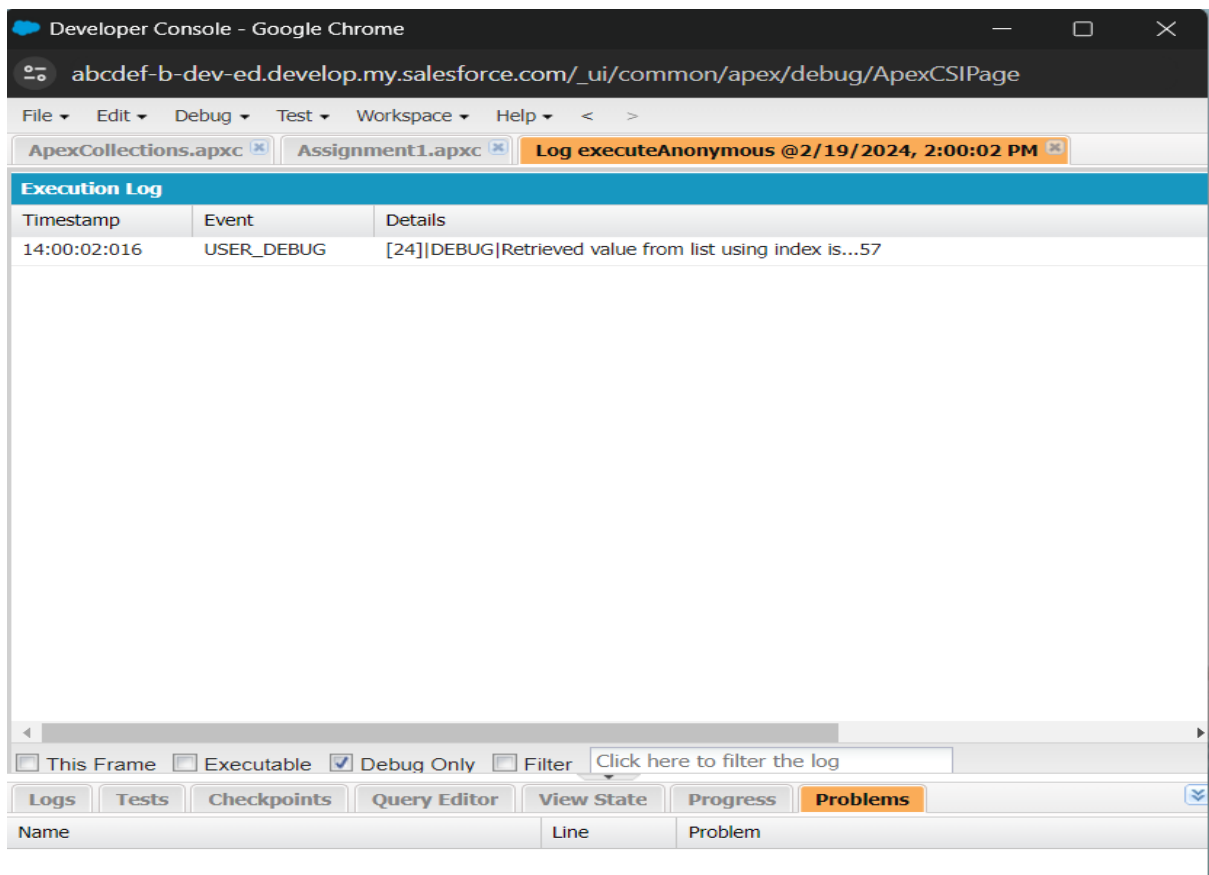
The screenshot shows the Salesforce Developer Console with the Execution Log for Assignment1.apxc. The log displays the following events:

Timestamp	Event	Details
13:05:48:014	USER_DEBUG	[10] DEBUG ...9
13:05:48:014	USER_DEBUG	[10] DEBUG ...3
13:05:48:014	USER_DEBUG	[10] DEBUG ...6
13:05:48:014	USER_DEBUG	[10] DEBUG ...9
13:05:48:015	USER_DEBUG	[10] DEBUG ...7
13:05:48:015	USER_DEBUG	[10] DEBUG ...4
13:05:48:015	USER_DEBUG	[10] DEBUG ...6
13:05:48:015	USER_DEBUG	[10] DEBUG ...2
13:05:48:015	USER_DEBUG	[10] DEBUG ...4
13:05:48:015	USER_DEBUG	[10] DEBUG ...8

- Retrieve values from list using index.



```
10 }
11 for(integer a=0; a<10; a++){
12     system.debug('...' +li[a]);
13 }
14 }
15 }
16 public static void methodretriveList(){
17     List<Integer> myList = new List<Integer>();
18     myList.add(57);
19     myList.add(67);
20     myList.add(77);
21     myList.add(87);
22     Integer myNumber = myList.get(0);
23 }
24 system.debug( 'Retrieved value from list using index is...' +myNumber);
25 }
26 }
```



Developer Console - Google Chrome

abcdef-b-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

ApexCollections.apxc Assignment1.apxc Log executeAnonymous @2/19/2024, 2:00:02 PM

**Execution Log**

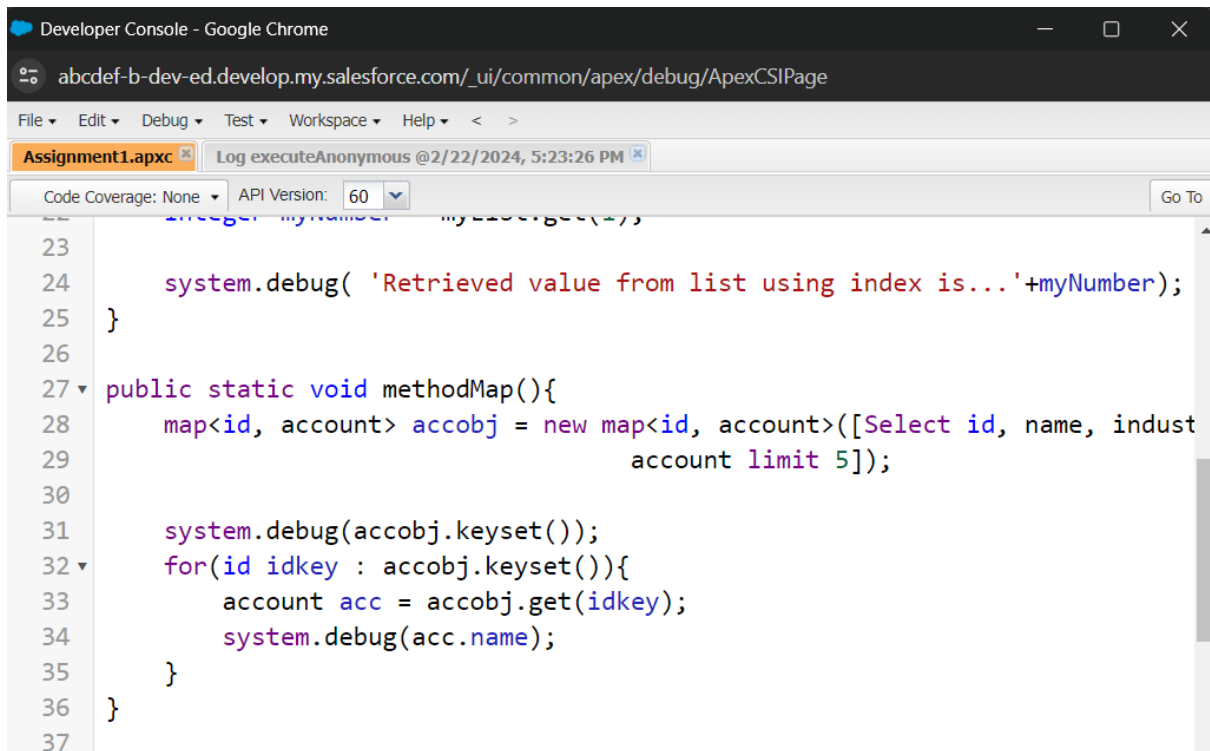
Timestamp	Event	Details
14:00:02:016	USER_DEBUG	[24] DEBUG Retrieved value from list using index is...57

☐ This Frame ☐ Executable ☒ Debug Only ☐ Filter

Logs Tests Checkpoints Query Editor View State Progress Problems

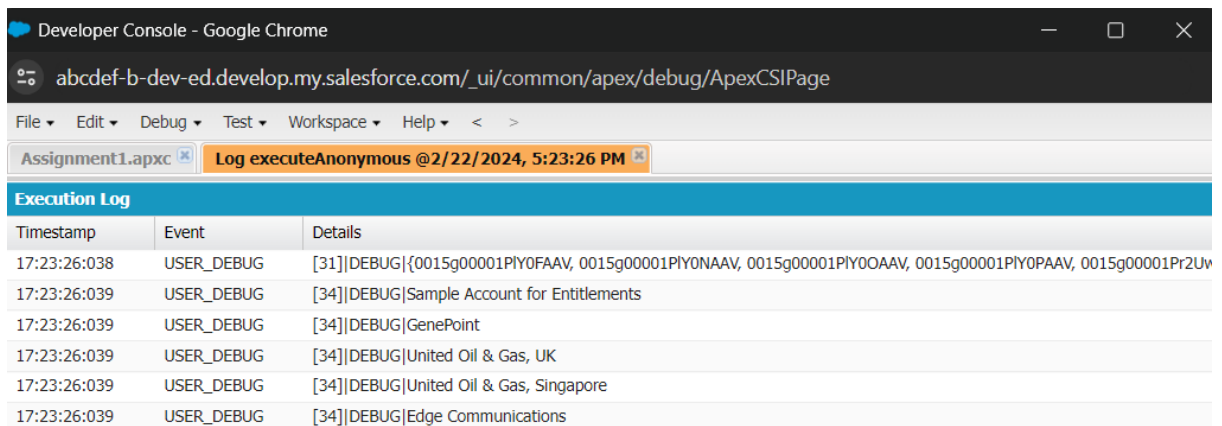
Name	Line	Problem
------	------	---------

- Create a map of Account Ids and Account objects.



```

23
24     system.debug( 'Retrieved value from list using index is...' + myNumber);
25 }
26
27 public static void methodMap(){
28     map<id, account> accobj = new map<id, account>([Select id, name, indust
29                                     account limit 5]);
30
31     system.debug(accobj.keySet());
32     for(id idkey : accobj.keySet()){
33         account acc = accobj.get(idkey);
34         system.debug(acc.name);
35     }
36 }
37
  
```



Timestamp	Event	Details
17:23:26:038	USER_DEBUG	[31] DEBUG {0015g00001PIY0FAAV, 0015g00001PIY0NAAV, 0015g00001PIY0OAAV, 0015g00001PIY0PAAV, 0015g00001Pr2Uv
17:23:26:039	USER_DEBUG	[34] DEBUG Sample Account for Entitlements
17:23:26:039	USER_DEBUG	[34] DEBUG GenePoint
17:23:26:039	USER_DEBUG	[34] DEBUG United Oil & Gas, UK
17:23:26:039	USER_DEBUG	[34] DEBUG United Oil & Gas, Singapore
17:23:26:039	USER_DEBUG	[34] DEBUG Edge Communications

- Write an example of aggregate functions without using a GROUP BY clause.

The screenshot shows the Salesforce Developer Console with the following details:

- Page Title:** Developer Console - Google Chrome
- URL:** abcdef-b-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage
- File:** Assignment1.apxc
- Log:** Log executeAnonymous @2/22/2024, 5:30:14 PM
- Execution Log:**

Timestamp	Event	Details
17:30:14:023	USER_DEBUG	[2]]DEBUG (AggregateResult:{expr0=87887777700})
- Enter Apex Code:**

```

1 list<aggregateResult> ar = [SELECT max(AnnualRevenue) from account];
2     system.debug(ar);

```

- Write an example Using SOQL Queries That Return One Record

The screenshot shows the Salesforce Developer Console with the following details:

- Page Title:** Developer Console - Google Chrome
- URL:** abcdef-b-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage
- File:** Assignment1.apxc
- Log:** Log executeAnonymous @2/22/2024, 5:39:36 PM
- Code Coverage:** None
- API Version:** 60
- Code:**

```

33         account acc = accobj.get(idkey);
34         system.debug(acc.name);
35     }
36 }
37
38 public static void methodLead(){
39     List<Lead> lea = New List<Lead>();
40     lea = [SELECT FirstName, LastName, Email from Lead Limit 1];
41     for (lead l : lea){
42         system.debug('FName..' + l.FirstName + 'LName..' + l.LastName + 'Email..');
43     }
44 }

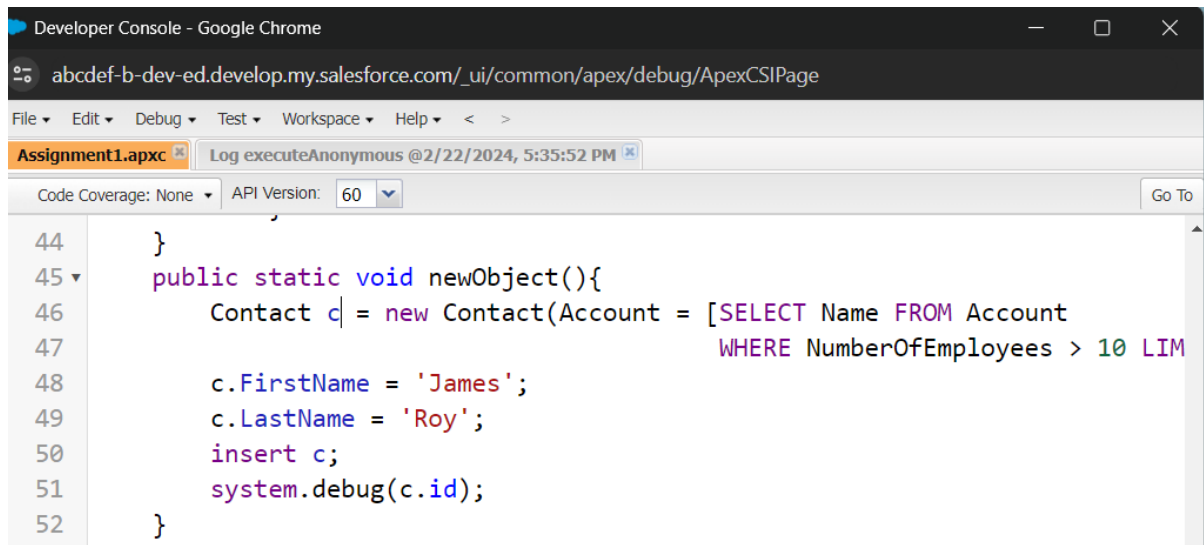
```

The screenshot shows the Salesforce Developer Console with the following details:

- Page Title:** Developer Console - Google Chrome
- URL:** abcdef-b-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage
- File:** Assignment1.apxc
- Log:** Log executeAnonymous @2/22/2024, 5:39:36 PM
- Execution Log:**

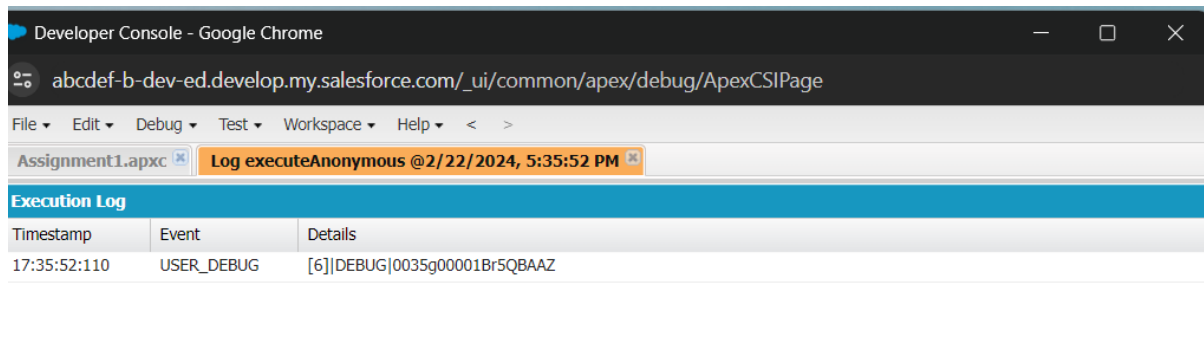
Timestamp	Event	Details
17:39:36:027	USER_DEBUG	[42]]DEBUG FName..BerthaLName..BoxerEmail..bertha@fcf.net

- Create new objects from SOQL queries on existing ones.



The screenshot shows the Salesforce Developer Console with the Apex code editor open. The code is for a class named `Assignment1.apxc` and is titled `Log executeAnonymous @2/22/2024, 5:35:52 PM`. The code defines a `newObject()` method that creates a new `Contact` object based on a SOQL query. The query selects the `Name` from the `Account` object where the `NumberOfEmployees` is greater than 10. The new `Contact` object is then created with the selected name, and its `FirstName` and `LastName` are set to 'James' and 'Roy' respectively. The object is then inserted into the database, and its ID is debugged.

```
44     }
45     public static void newObject(){
46         Contact c = new Contact(Account = [SELECT Name FROM Account
47                                         WHERE NumberOfEmployees > 10 LIM
48
49         c.FirstName = 'James';
50         c.LastName = 'Roy';
51         insert c;
52         system.debug(c.id);
53     }
```



The screenshot shows the Salesforce Developer Console with the Execution Log open. The log displays a single entry for a `USER_DEBUG` event, which is the debug statement from the previous code snippet. The log is titled `Execution Log` and has columns for `Timestamp`, `Event`, and `Details`.

Timestamp	Event	Details
17:35:52:110	USER_DEBUG	[6] DEBUG 0035g00001Br5QBAAZ