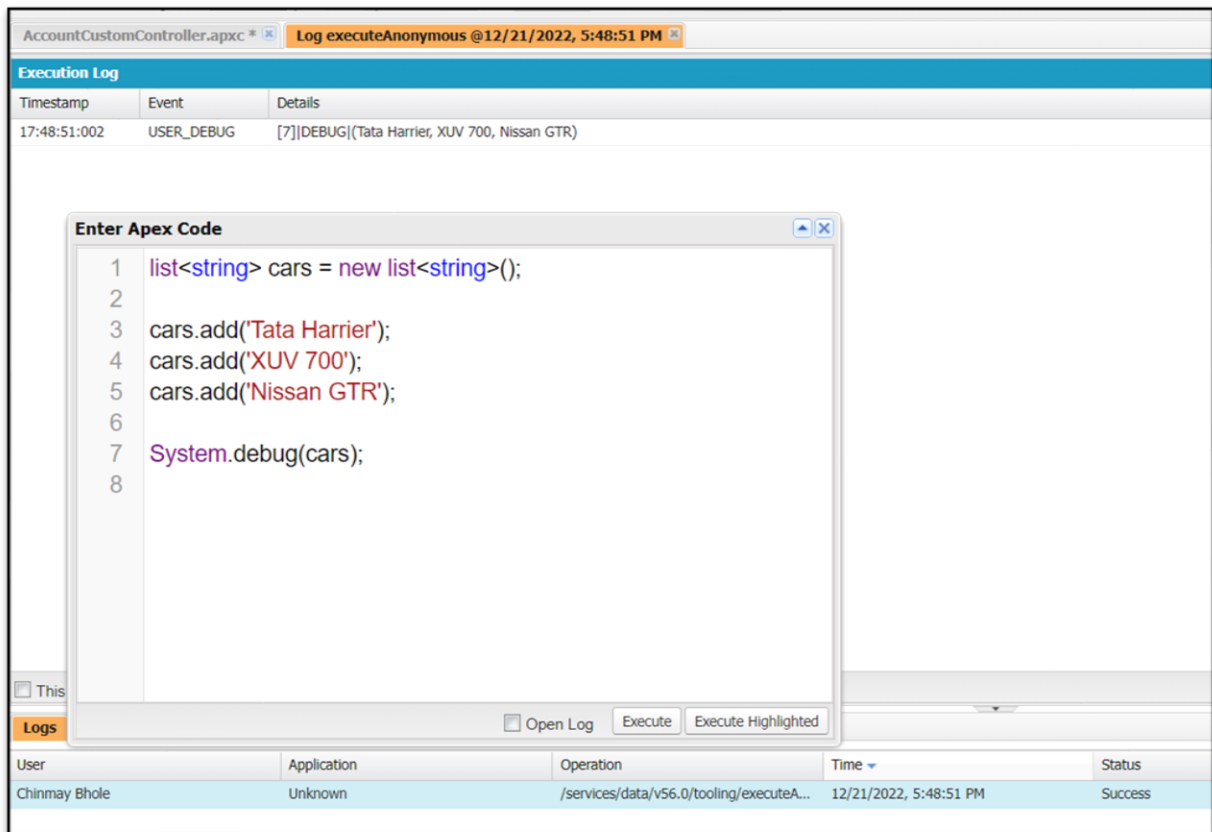


# Assignment 01

## Collection & SOQL Assignment

Q1. Add the Values to the list



The screenshot shows the Salesforce Developer Console. At the top, there are tabs for 'AccountCustomController.apxc' and 'Log executeAnonymous @12/21/2022, 5:48:51 PM'. Below the tabs is the 'Execution Log' section, which contains a table with the following data:

Timestamp	Event	Details
17:48:51:002	USER_DEBUG	[7][DEBUG](Tata Harrier, XUV 700, Nissan GTR)

Below the execution log is a window titled 'Enter Apex Code' containing the following code:

```
1 list<string> cars = new list<string>();
2
3 cars.add('Tata Harrier');
4 cars.add('XUV 700');
5 cars.add('Nissan GTR');
6
7 System.debug(cars);
8
```

At the bottom of the console, there is a 'Logs' section with a table showing the execution details:

User	Application	Operation	Time	Status
Chinmay Bhole	Unknown	/services/data/v56.0/tooling/executeA...	12/21/2022, 5:48:51 PM	Success

```
list<string> cars = new list<string>();
```

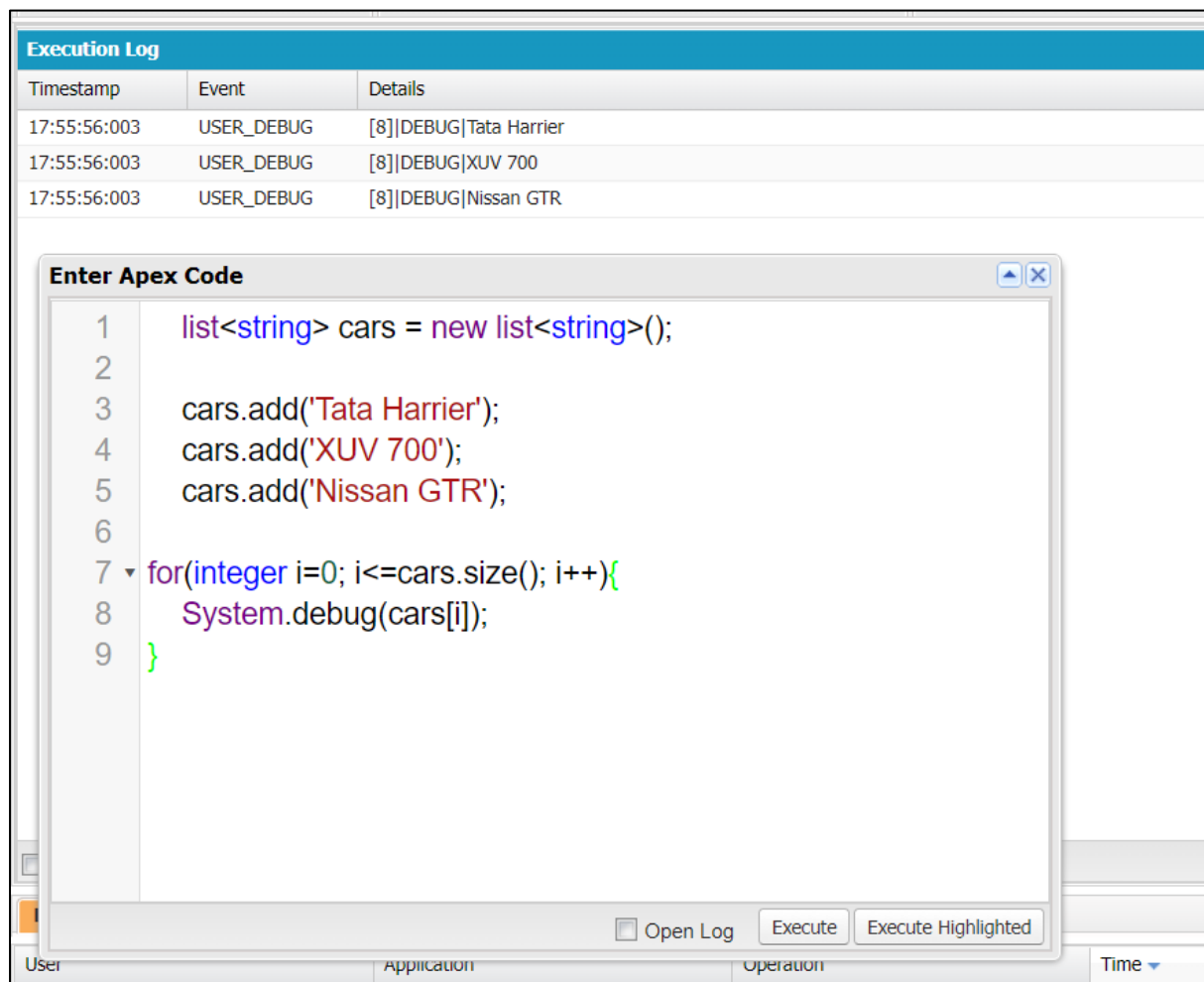
```
cars.add('Tata Harrier');
```

```
cars.add('XUV 700');
```

```
cars.add('Nissan GTR');
```

```
System.debug(cars);
```

**Q2.** Retrieve values from list using index.



The screenshot displays the Salesforce IDE interface. At the top, the 'Execution Log' window is open, showing three log entries:

Timestamp	Event	Details
17:55:56:003	USER_DEBUG	[8] DEBUG Tata Harrier
17:55:56:003	USER_DEBUG	[8] DEBUG XUV 700
17:55:56:003	USER_DEBUG	[8] DEBUG Nissan GTR

Below the log, the 'Enter Apex Code' window is open, showing the following code:

```
1 list<string> cars = new list<string>();
2
3 cars.add('Tata Harrier');
4 cars.add('XUV 700');
5 cars.add('Nissan GTR');
6
7 for(integer i=0; i<=cars.size(); i++){
8     System.debug(cars[i]);
9 }
```

At the bottom of the IDE, there are buttons for 'Open Log', 'Execute', and 'Execute Highlighted'. The bottom status bar shows 'User', 'Application', 'Operation', and 'Time'.

```
list<string> cars = new list<string>();
```

```
cars.add('Tata Harrier');
```

```
cars.add('XUV 700');
```

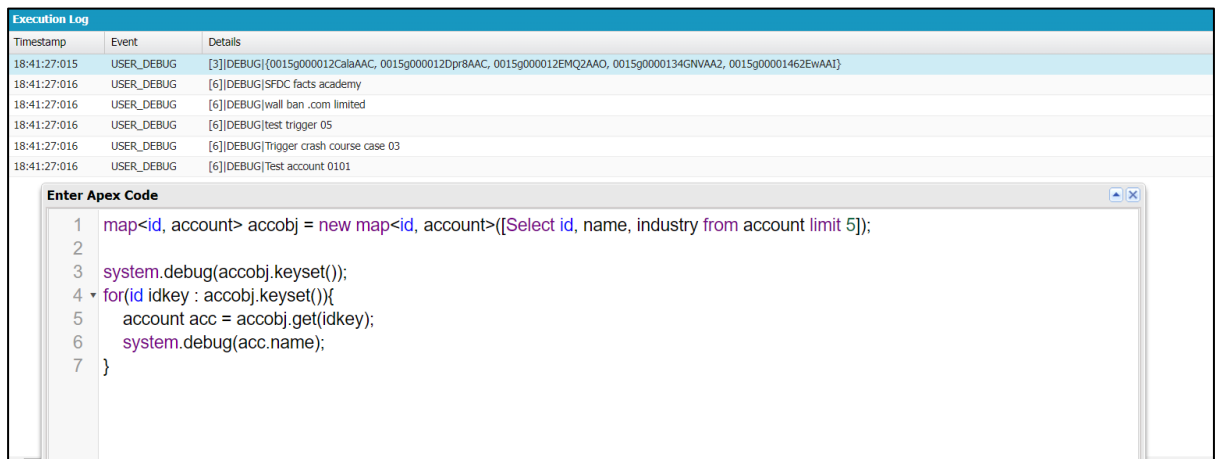
```
cars.add('Nissan GTR');
```

```
for(integer i=0; i<=cars.size(); i++){
```

```
    System.debug(cars[i]);
```

```
}
```

### Q3. Create a map of Account IDs and Account objects



The screenshot shows the Salesforce IDE interface. At the top, the 'Execution Log' is open, displaying a table with columns: Timestamp, Event, and Details. The log contains several entries, including a debug statement that prints a list of account IDs. Below the log, the 'Enter Apex Code' window is open, showing the following code:

```
1 map<id, account> accobj = new map<id, account>([Select id, name, industry from account limit 5]);
2
3 system.debug(accobj.keySet());
4 for(id idkey : accobj.keySet()){
5     account acc = accobj.get(idkey);
6     system.debug(acc.name);
7 }
```

```
map<id, account> accobj = new map<id, account>([Select id, name, industry from
account limit 5]);
```

```
system.debug(accobj.keySet());
```

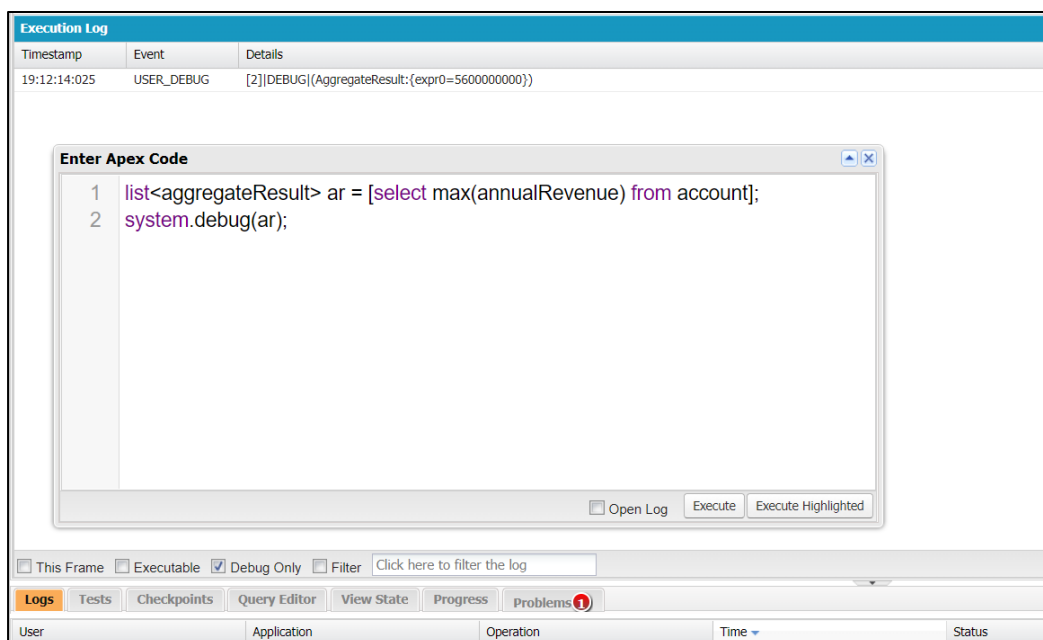
```
for(id idkey : accobj.keySet()){
```

```
    account acc = accobj.get(idkey);
```

```
    system.debug(acc.name);
```

```
}
```

### Q4. Write an example of Aggregate function without using a GROUP BY clause



The screenshot shows the Salesforce IDE interface. At the top, the 'Execution Log' is open, displaying a table with columns: Timestamp, Event, and Details. The log contains a single entry showing the result of an aggregate function. Below the log, the 'Enter Apex Code' window is open, showing the following code:

```
1 list<aggregateResult> ar = [select max(annualRevenue) from account];
2 system.debug(ar);
```

At the bottom of the IDE, there are tabs for 'Logs', 'Tests', 'Checkpoints', 'Query Editor', 'View State', 'Progress', and 'Problems'. The 'Logs' tab is selected, and the 'Execution Log' is visible. The log shows the result of the aggregate function: [2] | DEBUG | (AggregateResult: {expr0=5600000000}).

Q5. Write an example using SOQL queries that return one record

The screenshot shows the Salesforce IDE interface. At the top is the 'Execution Log' window with a table containing one entry:

Timestamp	Event	Details
19:15:21:013	USER_DEBUG	[3]   DEBUG   (Account: {Name=SFDC facts academy, Industry=Agriculture, Id=0015g00001462EwAAI})

Below the log is the 'Enter Apex Code' editor window. It contains the following code:

```
1 list<account> acc = [select name, industry from account where
2                       annualrevenue>100000 limit 1];
3 system.debug(acc);
```

At the bottom of the editor are buttons for 'Open Log', 'Execute', and 'Execute Highlighted'. Below the editor is a filter bar with checkboxes for 'This Frame', 'Executable', 'Debug Only' (checked), and 'Filter', followed by a link 'Click here to filter the log'.

Q6. CREATE new objects from SOQL queries on existing ones.

The screenshot shows the Salesforce IDE interface. At the top is the 'Execution Log' window, which is currently empty. Below it is the 'Enter Apex Code' editor window. It contains the following code:

```
1 contact con = new contact (account = [select name from account where
2                                       numberOfEmployees>10 limit 1];
3
4 con.firstname='John';
5 con.lastname='Wick';
6 insert con;
7 system.debug(con.id);
```

At the bottom of the editor are buttons for 'Open Log', 'Execute', and 'Execute Highlighted'.