

## Collection and SOQL Assignment

### 1> Add the values to the list

Ans-

```
List<string> Names = new list<string> ();
```

```
names.add('Veena');  
names.add('Pratik');  
names.add('Sakshi');  
names.add('Krishna');  
names.add('Archana');
```

```
system.debug('Total names =' + Names);
```

The screenshot displays the Salesforce IDE interface. At the top, the 'ListDemo.apxc' file is open, showing the following code:

```
1 public class ListDemo {  
2  
3     public void ListMethod() {  
4         List<string> List1 = new List<string>();  
5  
6         List1.add('Veena');  
7         List1.add('Pratik');  
8         List1.add('Sakshi');  
9         List1.add('Krishna');  
10        List1.add('Archana');  
11  
12        system.debug('Names=' + List1);  
13    }  
14 }
```

Below the code editor, the 'Problems' tab is active, showing the 'Apex Class Detail' for 'ListDemo'. The details include:

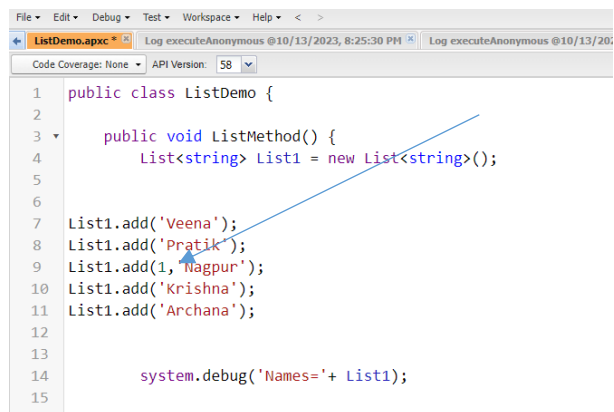
- Name: ListDemo
- Namespace Prefix:
- Status: Active
- Code Coverage: 0% (No coverage data)
- Created By: Veena G. 10/13/2023, 8:19 PM
- Last Modified By: Veena G. 10/13/2023, 8:19 PM

At the bottom, the 'Execution Log' is visible, showing a debug message:

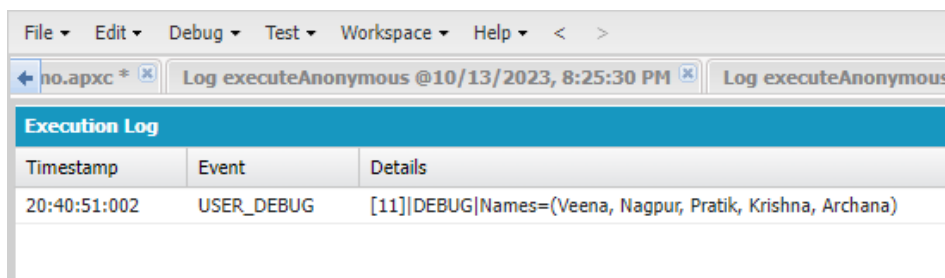
Timestamp	Event	Details
18:52:47:002	USER_DEBUG	[9] DEBUG Total names =(Veena, Pratik, Sakshi, Krishna, Archana)

2> Retrieve values from list using  
Ans- Add index as per below

index.

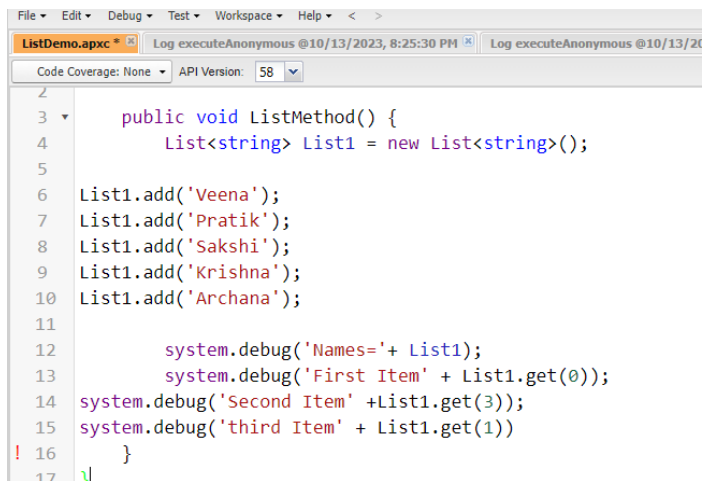


```
1 public class ListDemo {  
2  
3     public void ListMethod() {  
4         List<string> List1 = new List<string>();  
5  
6  
7         List1.add('Veena');  
8         List1.add('Pratik');  
9         List1.add(1, 'Nagpur');  
10        List1.add('Krishna');  
11        List1.add('Archana');  
12  
13  
14        system.debug('Names=' + List1);  
15    }  
16 }
```



Execution Log		
Timestamp	Event	Details
20:40:51:002	USER_DEBUG	[11] DEBUG Names=(Veena, Nagpur, Pratik, Krishna, Archana)

Another way to use index-



```
2  
3     public void ListMethod() {  
4         List<string> List1 = new List<string>();  
5  
6         List1.add('Veena');  
7         List1.add('Pratik');  
8         List1.add('Sakshi');  
9         List1.add('Krishna');  
10        List1.add('Archana');  
11  
12        system.debug('Names=' + List1);  
13        system.debug('First Item' + List1.get(0));  
14        system.debug('Second Item' + List1.get(3));  
15        system.debug('third Item' + List1.get(1))  
16    }  
17 }
```

ListDemo.apxc *	Log executeAnonymous @10/13/2023, 8:25:30 PM	Log executeAnonymous
Execution Log		
Timestamp	Event	Details
20:36:18:002	USER_DEBUG	[9] DEBUG Names=(Veena, Pratik, Sakshi, Krishna, Archana)
20:36:18:002	USER_DEBUG	[10] DEBUG First ItemVeena
20:36:18:002	USER_DEBUG	[11] DEBUG Second ItemKrishna
20:36:18:002	USER_DEBUG	[12] DEBUG third ItemPratik

### 3> Create a map of Account Ids and Account objects.

Ans-

Syntax----

```
public class DemoMap {
    public void DemoMapMethod(){

        Map<Id, Account> IdToAccountMap = new Map<Id, Account>();

        Account account1 = New Account(Name = 'Account 1', Industry = 'IT');
        Account account2 = New Account (Name = 'Account 2', Industry = ' HR');
        Account account3 = New Account (Name = 'Account 3', Industry ='Finance');

        IdToAccountMap.put(account1.Id, account1);
        IdToAccountMap.put(account2.Id, account2);
        IdToAccountMap.put(account3.Id, account3);

        for (Id accountId : IdToAccountMap.keySet()) {
            Account account = IdToAccountMap.get(accountId);
            System.debug('Company Name-' + account.Name);
            System.debug('Industry Name-' + account.Industry);
        }
    }
}
```

Result-

Execution Log		
Timestamp	Event	Details
13:42:24:004	USER_DEBUG	[13] DEBUG Company Name-Account 3
13:42:24:004	USER_DEBUG	[14] DEBUG Industry Name-Finance

### 4> Write an example of aggregate functions without using a GROUP BY clause.

Ans—

```
List<Double> numericValues = new List<Double>{10.5, 20.0, 15.5, 5.0, 30.0};
```

```
Double sum = 0;  
for (Double value : numericValues) {  
    sum += value;
```

```
System.debug('Sum-' + sum);  
}
```

Execution Log		
Timestamp	Event	Details
14:03:26:004	USER_DEBUG	[8] DEBUG Sum-10.5
14:03:26:004	USER_DEBUG	[8] DEBUG Sum-30.5
14:03:26:004	USER_DEBUG	[8] DEBUG Sum-46.0
14:03:26:005	USER_DEBUG	[8] DEBUG Sum-51.0
14:03:26:005	USER_DEBUG	[8] DEBUG Sum-81.0

For **average**-

```
Double average= numericValues.isEmpty() ? 0.0: sum/numericValues.size();
```

```
system.debug('Average-'+ average);  
}
```

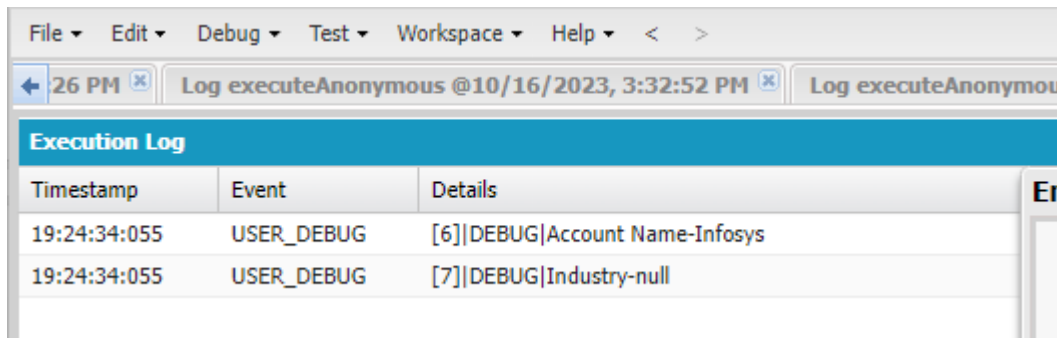
Execution Log		
Timestamp	Event	Details
15:37:46:003	USER_DEBUG	[8] DEBUG Sum-10.5
15:37:46:004	USER_DEBUG	[12] DEBUG Average-2.1
15:37:46:004	USER_DEBUG	[8] DEBUG Sum-30.5
15:37:46:004	USER_DEBUG	[12] DEBUG Average-6.1
15:37:46:004	USER_DEBUG	[8] DEBUG Sum-46.0
15:37:46:004	USER_DEBUG	[12] DEBUG Average-9.2
15:37:46:004	USER_DEBUG	[8] DEBUG Sum-51.0
15:37:46:004	USER_DEBUG	[12] DEBUG Average-10.2
15:37:46:004	USER_DEBUG	[8] DEBUG Sum-81.0
15:37:46:005	USER_DEBUG	[12] DEBUG Average-16.2

## 5> Write an example Using SOQL Queries That Return One Record

Ans- string accountIdtoQuery = '0012t00000X8nrLAAR';

```
Account singleAccount = [SELECT Id, Name, Industry From Account Where ID= :accountIdToQuery  
LIMIT 1];
```

```
if (singleAccount != Null) {  
    system.debug('Account Name-' + singleAccount.Name);  
    system.debug('Industry-' + singleAccount.Industry);  
}else {  
    system.debug('Account not found with ID-' + accountIdToQuery);  
}
```



The screenshot shows the 'Execution Log' window in the Salesforce IDE. It has a menu bar (File, Edit, Debug, Test, Workspace, Help) and a toolbar with a back arrow, a timestamp '26 PM', and two tabs for 'Log executeAnonymous @10/16/2023, 3:32:52 PM'. The log table contains the following data:

Timestamp	Event	Details
19:24:34:055	USER_DEBUG	[6] DEBUG Account Name-Infosys
19:24:34:055	USER_DEBUG	[7] DEBUG Industry-null

#### 6> Create new objects from SOQL queries on existing ones.

Ans---

```
List<Contact> contacts = [SELECT ID, Name, Email, AccountId From Contact where AccountId != Null  
LIMIT 5];
```

```
List<Opportunity> newOpportunities = new List<Opportunity>();
```

```
For (Contact contact : contacts){  
    Opportunity opp = new opportunity();  
    opp.Name = 'Opportunity for-' + contact.Name;
```

```
    opp.CloseDate = Date.today() + 30;
```

```
    opp.AccountId = contact.AccountId;
```

```

opp.Contact__c = contact.Id;
newOpportunities.add(opp);

system.debug('New Opportunity is created');
system.debug(' No contact record found for creating opportunities ');

}

```

39 PM Log executeAnonymous @10/16/2023, 3:37:46 PM Log executeAnonymous @10/16/2023, 3:37:46 PM		
Execution Log		
Timestamp	Event	Details
20:15:28:051	USER_DEBUG	[14] DEBUG New Opportunity is created
20:15:28:051	USER_DEBUG	[15] DEBUG  No contact record found for creating opportunities
20:15:28:053	USER_DEBUG	[14] DEBUG New Opportunity is created
20:15:28:053	USER_DEBUG	[15] DEBUG  No contact record found for creating opportunities
20:15:28:054	USER_DEBUG	[14] DEBUG New Opportunity is created
20:15:28:054	USER_DEBUG	[15] DEBUG  No contact record found for creating opportunities
20:15:28:055	USER_DEBUG	[14] DEBUG New Opportunity is created
20:15:28:055	USER_DEBUG	[15] DEBUG  No contact record found for creating opportunities
20:15:28:056	USER_DEBUG	[14] DEBUG New Opportunity is created
20:15:28:056	USER_DEBUG	[15] DEBUG  No contact record found for creating opportunities