**Project Development Phase**

### Exception Handling

|  |  |
| --- | --- |
| **NM ID** | **8073B6FD2C609D2A026419D64F1FD32** |
| **PROJECT TITTLE** | **BUILD IN EVENT MANAGEMENT USING SALESFORCE** |

## Exception Handling

Exception (fault) is an abnormal event that occurs generally after the execution of a program. It affects the normal program flow.  
→It is run time error through which execution is stopped  
→ Another name for run time error is exception error, there are two types of errors

* Syntax error (or) Compile time
* Runtime error (or) exception

→ A compile time (syntax) error occurs whenever we violate the syntax of the language  
→ The only way to overcome this scenario is by changing the source code  
→ In case of run time error this might occur not due to programming error but because of run time  
→ Environment such as

* Improper limits
* Lack of resources // insufficient memory
* Lack of connectivity

→ Such errors also stop the execution  
→ In order to overcome run time errors and to continue with the execution of the programming language a mechanism is provided known as exception handling

### Collections

It stores large volumes of information, there are three types of collections in sales force

* List
* Sat
* Map

**List:**  
A collection of similar volumes  
It is also re sizable

**Set:**  
Provides values which are unique  
It is also re sizable

**Map:**  
Collection of pairs  
Every pair is associated with a key that is associated with a value  
Pair →  
Key → access the value  –>(Can be an integer, string, id only )  
Value can be any type

**Syntax For List:**

Syntax

Template class, here we have to fix the data type at run time

**Creation Of List:**

* List ();
* Stud \_ names.sige à count of values existing in the list

**Input to the list:**

* Stud\_names. Add (‘Ashok’);
* Stud\_names. Add (‘Pandu’)
* Stud\_names.remove (‘Ashok’);

Remove → clear only one record from the list

* Stud\_names.clear ();

Clear (); → content of the list will be cleared

* Stud\_ names. Add (i1)

                ↓  
Removes common values of stud\_names and list:

**Output From The List:**

There are four ways to display  
1.For loop  
2.While loop  
3.Do- while  
4.For – each

**For:**  
For ( integer IND = 0; IND < stud\_names. Size ;  
IND ++)  
{  
System.debug(stud\_names [IND]);  
}  
In apex, we will work only for – each loop

**Syntax:**  
For each  
For (data type var\_name : collection \_name)  
{  
System. Debug (var\_name);  
}  
**Eg:**  
For (string Sn: stud\_names ) {  
System.debug (Sn);  
}  
Means each string in that collection has to be displayed  
Here, no need to specify starting & ending number and incrimination

**Map:**  
Put (key, value) → for adding  
Get (key) → for retrieving  
List → we can work only for ‘subject’  
Set → can’t work for subject this can work for primitive data types

 ↓  
If we have an ID, we can access any record

* We can’t create table through Apex to the DB
* We can’t drag table through Apex to the DB
* We can’t alter a table through Apex to the DB
* These statements are called DDL statements
* We can’t apply DDL statements through Apex
* We can give DML operations as input through Apex to the data base

Combine the tables

We can undelete a data from the sales force within a span of 45 days.

Standard tables

**SOQL:**  
Sales force object query language (SOQL) is used to retrieve the data by using “select” stmt  
In SOQL search only for table

**SOSL:**  
Sales force object search language (SOSL) is used to retrieve the data from the database, for this we used “find” stmt  
Here to search the entire database  
Select work with records (Table)  
Find work with text based search across from the DB returns field only but not records

**Save Point:**

We’ve to set ‘save point’ to particular points and not for all

* Most recently saved will come out first
* Max save points are 10

**Governor limits:**

* Governor limits are the rules which enforced restrictions at run time with respect to organizations

**Limits:**

* No. of SOQL queries : 100
* No. of query rows :50000
* No. of SOSL queries : 20
* No. of DML statements :150
* DML rows : 10000
* No. of script statements: 200000
* Maximum heap size : 3000000
* No. of callouts : 10
* No. of email invocations :10
* No. of field describes : 100
* No. of record type describes : 100
* No. of child relationships : 100
* No. of PICK lists, describes : 100
* No. of future calls : 10

**Collection Program:**

Public with sharing class call test 1 {  
Public static test method void main () {  
List < string > stud\_names = new list < string > ();  
Stud\_names. Add (‘Ashok’);  
Stud\_names. Add (‘Yadav’);  
Stud\_names. Add (‘Buddi’);  
For (string str: stud\_names) {  
System. Debug (‘name; ‘ +str);  
}  
List prog\_books =new list ();  
Prog\_books = « \_\_\_\_\_\_ »;  
Book\_c Sb = new book\_c ();  
Sb.name = ‘apex prog’;  
Sb.price \_ c = 200;  
Sb. Author\_c = ‘krishna’;  
Try {  
Insert prog\_books;  
}  
Catch (exception e) {  
System.debug (‘exceeded limit’);  
}  
Prog-books = « \_\_\_\_\_\_ »;  
For (Book\_c b: prog\_books){  
System.debug (b. price\_c);  
System.debug (b. author\_c);  
}  
}  
}