

# ***SPRINT EDUCATION***

*By Kashmira Merchant*



*Please view this  
slideshow in Microsoft  
PowerPoint as some  
slides contain GIFs  
which may appear static  
if viewed in MS Teams.  
Thank you 😊*



*Create a new org.*

*Renamed Contact object to Student.*

The screenshot displays the Salesforce Setup interface. At the top, there is a search bar labeled 'Search Setup' and a navigation bar with 'Setup', 'Home', and 'Object Manager'. The 'Object Manager' dropdown is open, showing 'Student' as the selected object. Below this, the 'Details' tab is active in the left sidebar. The main content area shows the 'Details' for the 'Student' object, including fields like 'Singular Label' (Student) and 'Plural Label' (Students), which are highlighted with a yellow box. Other settings like 'API Name', 'Contact', 'Custom', 'Enable Reports', 'Track Activities', 'Track Field History', 'Deployment Status', 'Help Settings', and 'Standard salesforce.com Help Window' are also visible.

Search Setup

Setup Home Object Manager

SETUP > OBJECT MANAGER  
**Student**

**Details**

- Fields & Relationships
- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Hierarchy Columns
- Scoping Rules
- Triggers
- Flow Triggers
- Validation Rules

**Details**

Description

API Name  
Contact  
Custom

Singular Label  
Student

Plural Label  
Students

Enable Reports

Track Activities

Track Field History

Deployment Status

Help Settings  
Standard salesforce.com Help Window

2

*Create new fields.*

*Added a Prior Qualification field that takes one of three values:*

- *Below Tertiary*
- *Tertiary Graduate*
- *Tertiary Post-Graduate*

*Marked the field as "Required".*

The screenshot shows the Salesforce Setup interface for the 'Student' object manager. The 'Contact Custom Field' 'Prior Qualification' is being configured. The field is marked as 'Required' and has three values: 'Below Tertiary', 'Tertiary Graduate', and 'Tertiary Post-Graduate'. The 'Required' checkbox is highlighted with a yellow box. The 'Values' table at the bottom is also highlighted with a yellow box.

**Setup > OBJECT MANAGER**  
**Student**

**Contact Custom Field**  
**Prior Qualification**  
[Back to Contact Fields](#) [Validation Rules \[0\]](#) [Help for this Page](#)

**Custom Field Definition Detail** [Edit](#) [Set Field-Level Security](#) [View Field Accessibility](#) [Where is this used?](#)

**Field Information**

Field Label	Prior Qualification	Object Name	Contact
Field Name	Prior_Qualification	Data Type	Picklist
API Name	Prior_Qualification__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Kashmira Merchant, 08/11/2022, 01:36	Modified By	Kashmira Merchant, 08/11/2022, 01:36

**General Options**

**Required** ☒

**Default Value** [i](#)

**Picklist Options**

**Restrict picklist to the values defined in the value set** ☒

**Controlling Field** [\[New\]](#)

**Values** [New](#) [Reorder](#) [Replace](#) [Printable View](#) [Chart Colors](#) [Values Help](#) [Beta!](#)

Action	Values	API Name	Default	Chart Colors	Modified By
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Deactivate</a>	Below Tertiary	Below Tertiary	<input type="checkbox"/>	Assigned dynamically	Kashmira Merchant, 08/11/2022, 01:36
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Deactivate</a>	Tertiary Graduate	Tertiary Graduate	<input type="checkbox"/>	Assigned dynamically	Kashmira Merchant, 08/11/2022, 01:36
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Deactivate</a>	Tertiary Post-Graduate	Tertiary Post-Graduate	<input type="checkbox"/>	Assigned dynamically	Kashmira Merchant, 08/11/2022, 01:36

*Added “Enrolment Progression” field that takes one of three values:*

- *Advanced*
- *Ultimate*
- *Extra*

The screenshot shows the Salesforce Setup interface for the 'Student' object. The 'Enrolment Progression' custom field is being defined. The left sidebar lists various setup options, with 'Fields & Relationships' selected. The main content area displays the 'Custom Field Definition Detail' for 'Enrolment Progression'. The 'Field Information' section shows the field label, name, API name, and description. The 'General Options' section shows the field is required and has a default value. The 'Picklist Options' section shows the field is restricted to values defined in the value set. The 'Values' section shows a table of values for the field.

**Field Information**

Field Label	Enrolment Progression	Object Name	Contact
Field Name	Enrolment_Progression	Data Type	Picklist
API Name	Enrolment_Progression__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Kashmira Merchant, 08/11/2022, 01:38	Modified By	Kashmira Merchant, 10/11/2022, 01:47

**General Options**

Required ☐

Default Value

**Picklist Options**

Restrict picklist to the values defined in the value set ☐

Controlling Field

**Values**

Action	Values	API Name	Default	Chart Colors	Modified By
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Deactivate</a>	Advanced	Advanced	<input type="checkbox"/>	Assigned dynamically	Kashmira Merchant, 08/11/2022, 01:38
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Deactivate</a>	Extra	Extra	<input type="checkbox"/>	Assigned dynamically	Kashmira Merchant, 08/11/2022, 01:38
<a href="#">Edit</a>   <a href="#">Del</a>   <a href="#">Deactivate</a>	Ultimate	Ultimate	<input type="checkbox"/>	Assigned dynamically	Kashmira Merchant, 08/11/2022, 01:38

**Validation Rules**

[New](#)

[illegible]

To prevent unauthorized changes to the field's value, I marked the field as read-only and wrote a trigger to determine the value of the field programmatically based on Prior Qualification (see Q6).

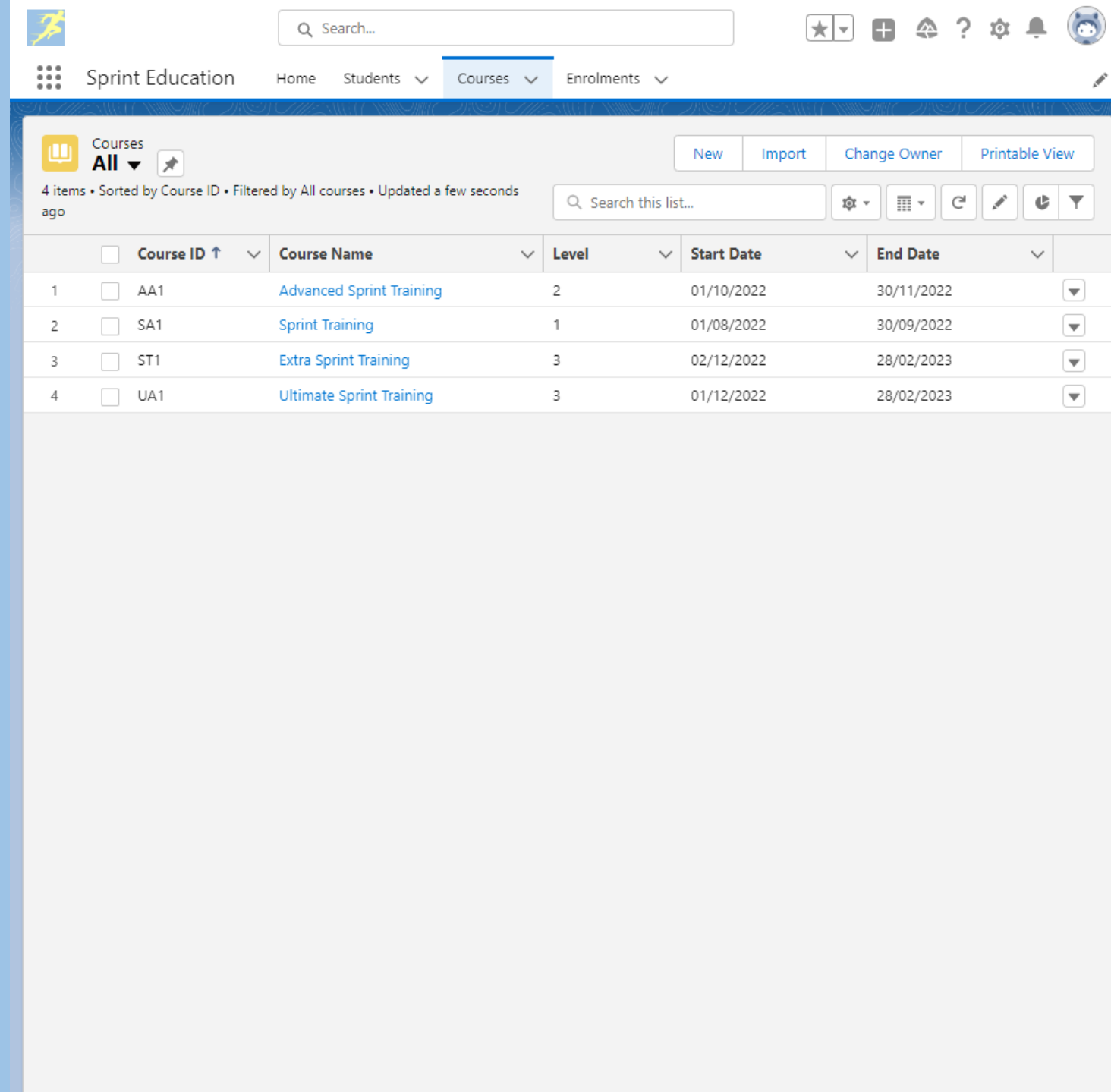
This way, the field's value is completely determined by back-end logic, which ensures a single source of truth.



# 3

*Create Course object.*


*Created a Course object  
and added a record for  
each course offered at  
Sprint.*










The screenshot displays the 'Sprint Education' web application. The top navigation bar includes a search bar, a user profile icon, and a menu with 'Home', 'Students', 'Courses', and 'Enrolments'. The 'Courses' section is active, showing a list of 4 items. The list is sorted by Course ID and filtered by All courses. The table below lists the courses with their IDs, names, levels, start dates, and end dates.


	<input type="checkbox"/>	Course ID ↑	Course Name	Level	Start Date	End Date	
1	<input type="checkbox"/>	AA1	<a href="#">Advanced Sprint Training</a>	2	01/10/2022	30/11/2022	▼
2	<input type="checkbox"/>	SA1	<a href="#">Sprint Training</a>	1	01/08/2022	30/09/2022	▼
3	<input type="checkbox"/>	ST1	<a href="#">Extra Sprint Training</a>	3	02/12/2022	28/02/2023	▼
4	<input type="checkbox"/>	UA1	<a href="#">Ultimate Sprint Training</a>	3	01/12/2022	28/02/2023	▼

*Changed locale from US to UK so that date fields are displayed in the dd/mm/yyyy format.*



Search...




 Sprint Education


Home


Students ▾

Courses ▾

Enrolments ▾



 Courses

All ▾ 

New







Import

Change Owner

Printable View

4 items • Sorted by Course ID • Filtered by All courses • Updated a few seconds ago

Search this list...



<input type="checkbox"/>	Course ID ↑ ▾	Course Name ▾	Level ▾	Start Date ▾	End Date ▾	
1	<input type="checkbox"/> AA1	<a href="#">Advanced Sprint Training</a>	2	01/10/2022	30/11/2022	▾
2	<input type="checkbox"/> SA1	<a href="#">Sprint Training</a>	1	01/08/2022	30/09/2022	▾
3	<input type="checkbox"/> ST1	<a href="#">Extra Sprint Training</a>	3	02/12/2022	28/02/2023	▾
4	<input type="checkbox"/> UA1	<a href="#">Ultimate Sprint Training</a>	3	01/12/2022	28/02/2023	▾

# 4

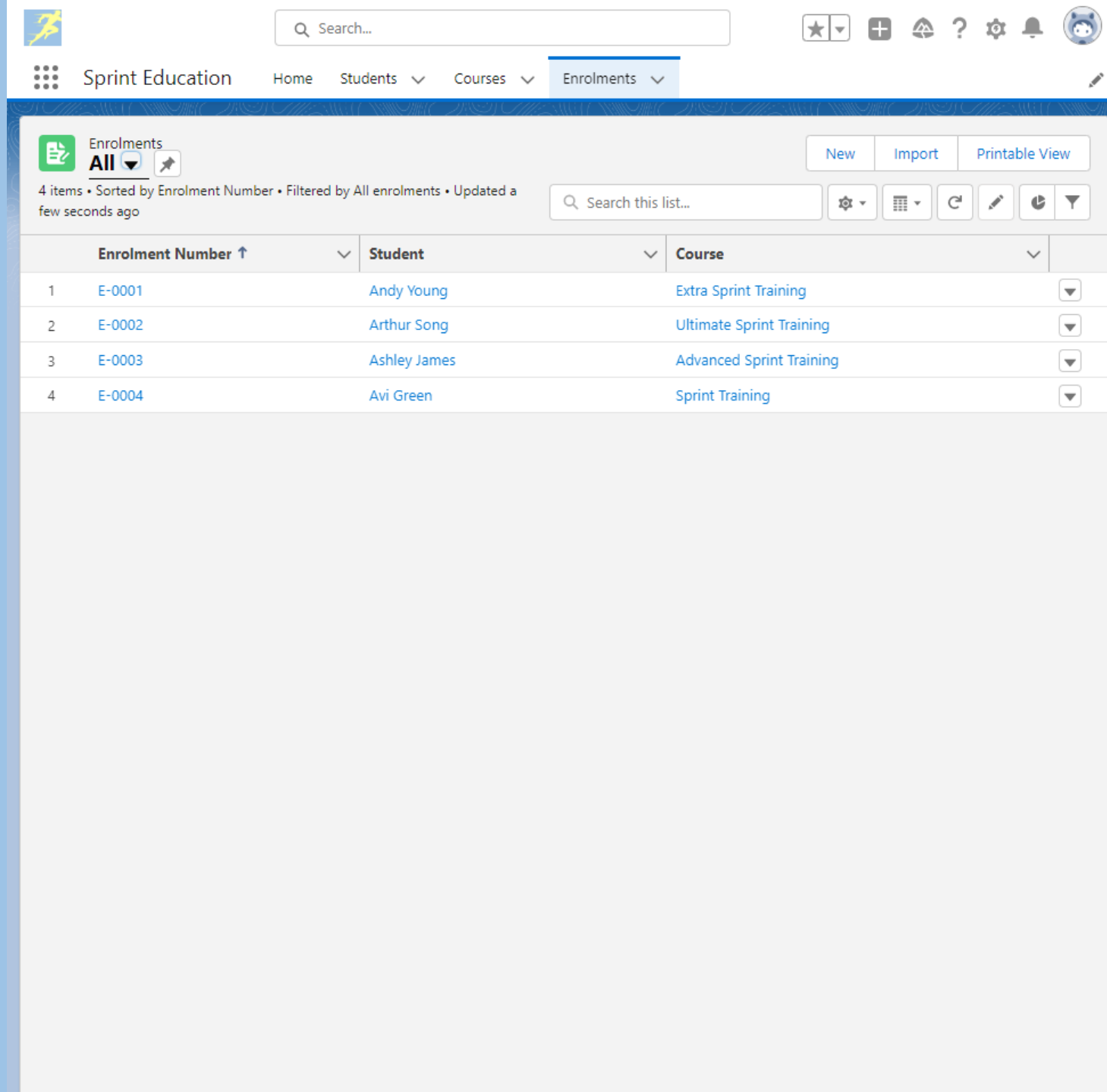
*Create junction object.*

*Created a junction object ("Enrolment") to model a two-way master-detail relationship between Students and Courses.*

The screenshot displays the Salesforce Setup interface. At the top, there's a navigation bar with 'Setup', 'Home', and 'Object Manager'. Below this, the 'Enrolment' object is selected under 'SETUP > OBJECT MANAGER'. The left sidebar shows a list of configuration options, with 'Fields & Relationships' highlighted. The main content area shows a table of fields for the 'Enrolment' object. The table has columns for 'FIELD LABEL', 'FIELD NAME', 'DATA TYPE', 'CONTROLLING FIELD', and 'INDEXED'. Five fields are listed: 'Course', 'Created By', 'Enrolment Number', 'Last Modified By', and 'Student'. The 'Course' and 'Student' rows are highlighted with orange boxes, indicating a master-detail relationship. The 'Course' row has a 'Master-Detail(Course)' data type, and the 'Student' row has a 'Master-Detail(Student)' data type. Both rows have a checkmark in the 'INDEXED' column and a dropdown arrow in the 'CONTROLLING FIELD' column.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Course	Course__c	Master-Detail(Course)		✓
Created By	CreatedById	Lookup(User)		
Enrolment Number	Name	Auto Number		✓
Last Modified By	LastModifiedById	Lookup(User)		
Student	Student__c	Master-Detail(Student)		✓

*List view of Enrolment  
junction object.*



The screenshot displays the 'Enrolments' section of the 'Sprint Education' application. The interface includes a top navigation bar with a search bar and various utility icons. Below this, a sub-header for 'Enrolments' shows a filter set to 'All' and a status of '4 items'. A secondary search bar and a set of action icons (settings, view, refresh, edit, print, filter) are also present. The main content is a table with three columns: 'Enrolment Number', 'Student', and 'Course'. The table lists four enrolments, each with a unique ID, a student name, and a course name. Each row has a dropdown arrow on the right for further actions.

	Enrolment Number ↑	Student	Course	
1	E-0001	Andy Young	Extra Sprint Training	▼
2	E-0002	Arthur Song	Ultimate Sprint Training	▼
3	E-0003	Ashley James	Advanced Sprint Training	▼
4	E-0004	Avi Green	Sprint Training	▼

# 6

*Test code for  
StudentInitialisation class*

*Unit test code for  
StudentInitialisation  
class.*

*Showing helper  
methods and unit test 1...*

```
force-app > main > default > classes > StudentInitialisationTest.cls > ...
1  /**
2   * @author Kashmira Merchant
3   * @description Unit tests for StudentInitialisation class (Question 6)
4   */
5  @isTest
   Run All Tests | Debug All Tests
6  private class StudentInitialisationTest {
7
8      /**
9       * @description Helper method to get student whose record Id matches studentId.
10      * @param studentId Record ID of the updated student.
11      */
12     private static Contact getStudent(String studentId) {
13
14         Contact student = [SELECT Id, Prior_Qualification__c, Enrolment_Progression__c
15                             FROM Contact
16                             WHERE Id =: studentId];
17
18         return student;
19     }
20
21     /**
22     * @description Unit test 1 - assert('Tertiary Post-Graduate', 'Advanced')
23     */
24     @isTest
   Run Test | Debug Test
25     private static void testAdvanced() {
26
27         Contact newStudent = new Contact();
28
29         newStudent.FirstName='Test1';
30         newStudent.LastName='Advanced';
31         newStudent.Prior_Qualification__c = 'Tertiary Post-Graduate';
32         newStudent.Enrolment_Progression__c = 'Advanced';
33
34         insert newStudent;
35
36         List<Contact> toUpdate = new List<Contact>{newStudent};
37         StudentInitialisation.updateEnrolmentProgHandler(toUpdate);
38
39         Contact updatedStudent = getStudent(newStudent.Id);
40
41         System.assertEquals('Advanced', newStudent.Enrolment_Progression__c);
42     }
43
44 }
45
```



## Showing unit tests 2 and 3.

```
force-app > main > default > classes > StudentInitialisationTest.cls > ...
46  /**
47   * @description Unit test 2 - assert('Tertiary Graduate', 'Ultimate')
48   */
49   @isTest
    Run Test | Debug Test
    private static void testUltimate(){
50
51       Contact newStudent = new Contact();
52
53       newStudent.FirstName='Test2';
54       newStudent.LastName='Ultimate';
55       newStudent.Prior_Qualification__c = 'Tertiary Graduate';
56       newStudent.Enrolment_Progression__c = 'Ultimate';
57
58       insert newStudent;
59
60       List<Contact> toUpdate = new List<Contact>{newStudent};
61       StudentInitialisation.updateEnrolmentProgHandler(toUpdate);
62
63       Contact updatedStudent = getStudent(newStudent.Id);
64
65       System.assertEquals('Ultimate', updatedStudent.Enrolment_Progression__c);
66
67   }
68
69   /**
70   * @description Unit test 3 - assert('Below Tertiary', 'Extra')
71   */
72   @isTest
    Run Test | Debug Test
    private static void testExtra(){
73
74       Contact newStudent = new Contact();
75
76       newStudent.FirstName='Test3';
77       newStudent.LastName='Extra';
78       newStudent.Prior_Qualification__c = 'Below Tertiary';
79       newStudent.Enrolment_Progression__c = 'Extra';
80
81       insert newStudent;
82
83       List<Contact> toUpdate = new List<Contact>{newStudent};
84       StudentInitialisation.updateEnrolmentProgHandler(toUpdate);
85
86       Contact updatedStudent = getStudent(newStudent.Id);
87
88       System.assertEquals('Extra', updatedStudent.Enrolment_Progression__c);
89
90   }
91
92   }
93
94
```

# 7

*StudentInitialisation class  
implementation*

*StudentInitialisation handles the initialization of the Prior Qualification and Enrolment Progression fields of current and future students.*

Using `Math.random()` to generate a random index into the `studentList` object.

`Math.random()` returns a number between 0 to 1, which is multiplied by the length of `studentList` to give a float between 0 and 2. `Math.floor()` rounds this down to the nearest integer. However, as this number is of type Decimal, it must be passed to `Integer.ValueOf()` to convert it to an Integer.

```
force-app > main > default > classes > StudentInitialisation.cls > ...
1  /**
2   * @author Kashmira Merchant
3   * @description Methods for initialising Prior Qualification and Enrolment Progression fields (Question 7).
4   */
5  public class StudentInitialisation {
6
7      /**
8       * @description Randomly assigns a prior qualification to each record in studentList.
9       * @param studentList List of Student records.
10      */
11     public static void assignPriorQual(List<Contact> studentList) {
12
13         List<String> priorQuals = new List<String>{'Below Tertiary',
14                                                  'Tertiary Graduate',
15                                                  'Tertiary Post-Graduate'};
16
17         for (Contact student : studentList){
18
19             // Generating a random index into priorQuals list
20             Integer i = Integer.valueOf(Math.floor(Math.random()*priorQuals.size()));
21
22             student.Prior_Qualification__c = priorQuals[i];
23
24         }
25
26         update studentList;
27     }
28
29     /**
30      * @description Determines a student's enrolment progression. Called by the updateEnrolmentProg trigger.
31      * @param studentList List of Student records.
32      */
33     public static void updateEnrolmentProgHandler(List<Contact> studentList) {
34
35         for (Contact student : studentList){
36
37             switch on student.Prior_Qualification__c { // Used switch in place of if-else for parsimony
38
39                 when 'Below Tertiary' {
40                     student.Enrolment_Progression__c = 'Advanced';
41                 }
42                 when 'Tertiary Graduate' {
43                     student.Enrolment_Progression__c = 'Ultimate';
44                 }
45                 when 'Tertiary Post-Graduate' {
46                     student.Enrolment_Progression__c = 'Extra';
47                 }
48                 when else {
49                     null;
50                 }
51             }
52
53         }
54     }
55 }
56
57 }
58
59 }
```

## *updateEnrolmentProgTrigger*

StudentInitialisation.updateEnrolmentProgHandler is called inside the updateEnrolProg trigger, which fires whenever a new student is added or an existing student's Prior Qualification is changed.

This ensures that students that Enrolment Progression is also populated for students that are added in the future.

```
force-app > main > default > triggers > updateEnrolmentProg.trigger > ...
1  /**
2   * @author Kashmira Merchant
3   * @description Trigger to update Enrolment Progression field of new and existing students.
4   * @param before Denotes before context. Need to modify Enrolment Progression
5   *               field in Trigger.new, which is only writable in before context.
6   * @param insert Denotes insert context. Ensures trigger is fired if a new student is added
7   * @param update Denotes update context. Ensures trigger is fired if existing
8   *               student's Prior Qualification is changed.
9   */
10 trigger updateEnrolmentProg on Contact (before insert, before update) {
11
12     StudentInitialisation.updateEnrolmentProgHandler(Trigger.new);
13
14 }
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
```

*Showing  
StudentInitialisation  
methods in action to  
populate Prior  
Qualification and  
Enrolment Progression  
fields.*

The image shows a development environment with an IDE at the top and a web application at the bottom.

**IDE (Top):** The IDE window displays several tabs: `StudentInitialisation.apxc`, `StudentInitialisationTest.apxc`, `addEnrolment.apxt`, `AddEnrolmentTest.apxc`, `TestVersion.apxc`, and `updateEnrolmentProg.a`. The `StudentInitialisation.apxc` tab is active, showing Apex code. A pop-up window titled "Enter Apex Code" is open, displaying the following code:

```
1 List<Contact> studentList = new List<Contact>([
2     SELECT Id,
3         Prior_Qualification__c,
4         Enrolment_Progression__c
5     FROM Contact
6 ]);
7 StudentInitialisation.assignPriorQual(studentList);
8
```

The IDE also shows a "Logs, Tests, and Problems" panel at the bottom.

**Web Application (Bottom):** The web application interface shows a "Students" section with a dropdown menu set to "All Students". There are buttons for "New", "Import", "Add to Campaign", "Send List Email", and "Printable View". Below these buttons, a table displays a list of students. The table has columns for "Name", "Enrolment Progression", and "Prior Qualification". A yellow rounded rectangle highlights the "Enrolment Progression" and "Prior Qualification" columns. The table contains 12 rows of student data.

	Name ↑	Enrolment Progression	Prior Qualification
1	<input type="checkbox"/> Andy Young		
2	<input type="checkbox"/> Arthur Song		
3	<input type="checkbox"/> Ashley James		
4	<input type="checkbox"/> Avi Green		
5	<input type="checkbox"/> Babara Levy		
6	<input type="checkbox"/> Edna Frank		
7	<input type="checkbox"/> Jack Rogers		
8	<input type="checkbox"/> Jake Llorrac		
9	<input type="checkbox"/> Jane Grey		
10	<input type="checkbox"/> John Bond		
11	<input type="checkbox"/> Josh Davis		
12	<input type="checkbox"/> Lauren Boyle		

## Test results

The screenshot shows an IDE with several tabs open: `StudentInitialisation.apxc`, `StudentInitialisationTest.apxc` (active), `addEnrolment.apxt`, `AddEnrolmentTest.apxc`, `TestVersion.apxc`, and `updateEnrolmentProg.a`. The `StudentInitialisationTest.apxc` tab displays the following code:

```
47  @description Umit test 2 - assert( 'Tertiary Graduate', 'Ultimate' )
48  */
49  @isTest
50  private static void testUltimate(){
51
52      Contact newStudent = new Contact();
53
54      newStudent.FirstName='Test2';
55      newStudent.LastName='Ultimate';
56      newStudent.Prior_Qualification__c = 'Tertiary Graduate';
57      newStudent.Enrolment_Progression__c = 'Ultimate';
58
59      insert newStudent;
60
61      List<Contact> toUpdate = new List<Contact>{newStudent};
62      StudentInitialisation.updateEnrolmentProgHandler(toUpdate);
63
64      Contact updatedStudent = getStudent(newStudent.Id);
65
66      System.assertEquals('Ultimate', updatedStudent.Enrolment_Progression__c);
67  }
68
69  /**
70  * @description Umit test 3 - assert('Below Tertiary', 'Extra')
71  */
72  @isTest
73  private static void testExtra(){
74
```

Below the code editor, the **Tests** tab is selected, showing a table of test results. A yellow box highlights the test results for `StudentInitialisationTest`.

Status	Test Run	Enqueued Time	Duration	Failures	Total
✗	TestRun @ 1:18:27 pm			2	3
✗	StudentInitialisationTest			2	3
✗	testAdvanced		0:00		
✗	testExtra		0:00		
✓	testUltimate		0:00		

To the right of the test results table, the **Overall Code Coverage** table is displayed:

Class	Percent	Lines
<b>Overall</b>	<b>71%</b>	
addEnrolment	100%	10/10
StudentInitialisation	52%	9/17
updateEnrolmentProg	100%	1/1

*Test coverage is less than 100% because we are only testing*  
`StudentInitialisation.updateEnrolProgHandler()`, *not*  
`StudentInitialisation.assignPriorQual()`.

The screenshot shows an IDE with several tabs open: `StudentInitialisation.apxc`, `StudentInitialisationTest.apxc` (active), `addEnrolment.apxt`, `AddEnrolmentTest.apxc`, `TestVersion.apxc`, and `updateEnrolmentProg.a`. The active tab displays Apex code for `StudentInitialisationTest`. The code includes a test method `testUltimate()` and a comment for `testExtra()`. The test results pane at the bottom shows a test run at 1:18:27 pm with the following results:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✗	TestRun @ 1:18:27 pm			2	3
✗	StudentInitialisationTest			2	3
✗	testAdvanced		0:00		
✗	testExtra		0:00		
✓	testUltimate		0:00		

An overall code coverage summary is displayed in a yellow-bordered box:

Overall Code Coverage		
Class	Percent	Lines
<b>Overall</b>	<b>71%</b>	
addEnrolment	100%	10/10
StudentInitialisation	52%	9/17
updateEnrolmentProg	100%	1/1

# 8

*AddEnrolment trigger test  
code*



## Initial trigger test code.

Line 23 queries production data. However, this is not allowed by default. It is possible to suppress this restriction by specifying `SeeAllData=True` in the `@isTest` declaration, however, this results in the code being tested becoming dependent on the data that is present in the org.

**SOLUTION:** The preferred approach is *create test data*, see next slide...

```
force-app > main > default > classes > AddEnrolmentTest1.cls > ...
1  /**
2   * @author Kashmira Merchant
3   * @description Test for AddEnrolment trigger (v1).
4   */
5   @isTest(SeeAllData=True)
6   Run All Tests | Debug All Tests
7   private class AddEnrolmentTest1 {
8       @isTest(SeeAllData=True)
9       Run All Tests | Debug All Tests
10      private static void testAddEnrolmentTrigger() {
11
12          // Create test data...
13          // ...dummy new Student
14          Contact newStudent = new Contact();
15
16          newStudent.FirstName = 'Harry';
17          newStudent.LastName = 'Potter';
18          newStudent.Prior_Qualification__c = 'Below Tertiary';
19          newStudent.Enrolment_Progression__c = 'Advanced';
20
21          insert newStudent;
22
23          // Query enrolment of newly inserted student
24          List<Enrolment__c> enrolmentList = [SELECT Id, Student__c
25                                             FROM Enrolment__c
26                                             WHERE Student__r.Id =: newStudent.Id];
27
28          // Check that enrolment was returned
29          System.assertNotEquals(enrolmentList.size(), 0);
30      }
31  }
32  }
```

## Improved version of trigger test.

This is my second (improved) version of the trigger. I removed `SeeAllData=true` and instead created a dummy SAI Course record. The dummy new student is then associated to this dummy SA1 and the test is evaluated on the resulting Enrolment record.

```
force-app > main > default > classes > AddEnrolmentTest2.cls > ...
1  /**
2   * @author Kashmira Merchant
3   * @description Test for AddEnrolment trigger (v2).
4   */
5  @isTest
6  Run All Tests | Debug All Tests
7  private class AddEnrolmentTest2 {
8
9      @isTest
10     Run Test | Debug Test
11     private static void testAddEnrolmentTrigger() {
12
13         // Create test data...
14         // ...dummy SA1
15         Course__c dummySA1 = new Course__c();
16
17         dummySA1.Name = 'Sprint Training';
18         dummySA1.Foundation_Course__c = true;
19         dummySA1.Level__c = 1;
20         dummySA1.Course_ID__c = 'SA1';
21
22         insert dummySA1;
23
24         // ...dummy new Student
25         Contact newStudent = new Contact();
26
27         newStudent.FirstName = 'Harry';
28         newStudent.LastName = 'Potter';
29         newStudent.Prior_Qualification__c = 'Below Tertiary';
30         newStudent.Enrolment_Progression__c = 'Advanced';
31
32         insert newStudent;
33
34         // Query enrolment of newly inserted student
35         List<Enrolment__c> enrolmentList = [SELECT Id, Student__c
36                                           FROM Enrolment__c
37                                           WHERE Student__r.Id =: newStudent.Id];
38
39         // Check that enrolment was returned
40         System.assertNotEquals(enrolmentList.size(), 0);
41     }
42 }
43
44
45
46
47
```

# 9

*AddEnrolment trigger test*

## Trigger to associate a new Student to the SA1 course.

Initially, I had hardcoded the record ID of the SA1 Course on line 24 (i.e. `newEnrolment.Course__c = 'a00Do000001ZjYxIAK'`). However, this creates a dependency on the actual value of the record ID.

In order to remove the dependency and make the code more general + robust, I created a new field on the Course object, called "Foundation Course" - a checkbox that takes the value `true` if a course is a foundational course that all new students must enrol in when they start at Sprint, e.g. SA1.




This generalizes the code – for each new student, I can now simply loop over the list of foundation courses and create the necessary Enrolment record for each foundation course. Also, no code needs to be changed if more foundation courses are added in the future.

```
force-app > main > default > triggers > addEnrolment.trigger > ...
1  /**
2   * @author Kashmira Merchant
3   * @description Creates an association (Enrolment record) between newly
4   *               created Student and each foundation course (SA1).
5   */
6  trigger addEnrolment on Contact (after insert) {
7
8      // List of Enrolments to store new Enrolment records
9      List<Enrolment__c> enrolmentList = New List<Enrolment__c>();
10
11     // Get list of foundation courses
12     List<Course__c> foundationCourses = [SELECT Id, Foundation_Course__c
13                                         FROM Course__c
14                                         WHERE Foundation_Course__c = true];
15
16     // For each newly created student...
17     for(Contact newStudent : Trigger.new){
18
19         // For each foundation course...
20         for (Course__c course : foundationCourses) {
21
22             // Create a new Enrolment record
23             Enrolment__c newEnrolment = new Enrolment__c();
24             newEnrolment.Course__c = course.Id;
25             newEnrolment.Student__c = newStudent.id;
26
27             // Add to list of Enrolments
28             enrolmentList.add(newEnrolment);
29         }
30     }
31
32     if(enrolmentList.size() > 0){
33
34         // Insert all new Enrolment records (bulkification)
35         insert enrolmentList;
36     }
37
38 }
39
40 }
41
42
43
44
45
46
47
```

Course  
**Sprint Training**

New Student Edit New Opportunity

Related **Details**

Course Name	Sprint Training	Owner	 <a href="#">Kashmira Merchant</a>
Course ID	SA1		
Level	1		
Start Date	01/08/2022		
End Date	30/09/2022		
	Foundation Course		
	<input checked="" type="checkbox"/>		
Created By	 <a href="#">Kashmira Merchant</a> 08/11/2022, 02:06	Last Modified By	 <a href="#">Kashmira Merchant</a> 09/11/2022, 00:50

**Activity**

New Event New Task Log a Call Email

Set up an event...

Add

Filters: All time • All activities • All types

Refresh • Expand All • View All

▼ Upcoming & Overdue

No activities to show.  
Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

View of the Details page for the SA1 course. Note the "Foundation Course" checkbox is ticked.

## Test results.

The screenshot shows an IDE with a code editor and a test results panel. The code editor displays a Java test class named `AddEnrolmentTest2` with a static method `testAddEnrolmentTrigger()`. The test method creates dummy data for a course and a student, then inserts them into the database. The test results panel at the bottom shows a successful test run for the `testAddEnrolmentTrigger` method, with a duration of 0:00, 0 failures, and 1 total line. The overall code coverage is 71%.

```
1 /**
2  * @author Kashmira Merchant
3  * @description Test for AddEnrolment trigger (v2).
4  */
5 @isTest
6 private class AddEnrolmentTest2 {
7
8     @isTest
9     private static void testAddEnrolmentTrigger() {
10
11         // Create test data...
12         // ...dummy SA1
13         Course__c dummySA1 = new Course__c();
14
15         dummySA1.Name = 'Sprint Training';
16         dummySA1.Foundation_Course__c = true;
17         dummySA1.Level__c = 1;
18         dummySA1.Course_ID__c = 'SA1';
19
20         insert dummySA1;
21
22         // ...dummy new Student
23         Contact newStudent = new Contact();
24
25         newStudent.FirstName = 'Harry';
26         newStudent.LastName = 'Potter';
27         newStudent.Prior_Qualification__c = 'Below Tertiary';
28         newStudent.Enrolment_Progression__c = 'Advanced';
29     }
```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 1:26:27 pm			0	1
✓	AddEnrolmentTest2			0	1
✓	testAddEnrolmentTrigger		0:00	0	1

Overall Code Coverage		
Class	Percent	Lines
<b>Overall</b>	<b>71%</b>	
addEnrolment	100%	10/10
StudentInitialisation	52%	9/17
updateEnrolmentProg	100%	1/1

## Test coverage.

The screenshot shows an IDE with several tabs open: `StudentInitialisation.apxc`, `StudentInitialisationTest.apxc`, `addEnrolment.apxt`, and `AddEnrolmentTest2.apxc`. The `AddEnrolmentTest2.apxc` tab is active, displaying the following Apex code:

```
1  /**
2   * @author Kashmira Merchant
3   * @description Test for AddEnrolment trigger (v2).
4   */
5  @isTest
6  private class AddEnrolmentTest2 {
7
8      @isTest
9      private static void testAddEnrolmentTrigger() {
10
11          // Create test data...
12          // ...dummy SA1
13          Course__c dummySA1 = new Course__c();
14
15          dummySA1.Name = 'Sprint Training';
16          dummySA1.Foundation_Course__c = true;
17          dummySA1.Level__c = 1;
18          dummySA1.Course_ID__c = 'SA1';
19
20          insert dummySA1;
21
22          // ...dummy new Student
23          Contact newStudent = new Contact();
24
25          newStudent.FirstName = 'Harry';
26          newStudent.LastName = 'Potter';
27          newStudent.Prior_Qualification__c = 'Below Tertiary';
28          newStudent.Enrolment_Progression__c = 'Advanced';
29      }
```

Below the code editor, the **Tests** tab is selected, showing a table of test results:

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 1:26:27 pm			0	1
✓	AddEnrolmentTest2			0	1
✓	testAddEnrolmentTrigger		0:00		

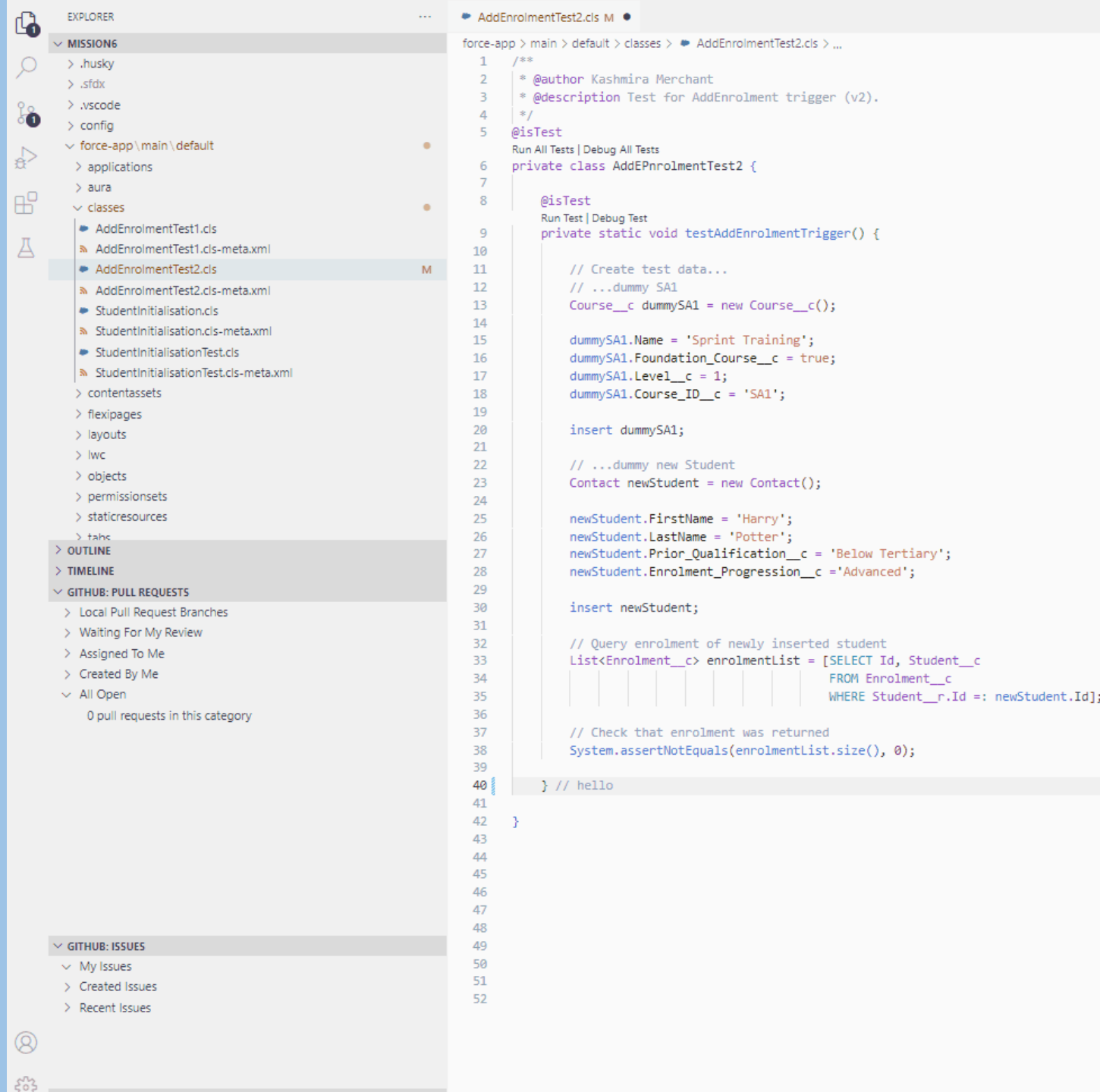
On the right side, the **Overall Code Coverage** panel is displayed, showing the following data:

Class	Percent	Lines
<b>Overall</b>	<b>71%</b>	
addEnrolment	100%	10/10
StudentInitialisation	52%	9/17
updateEnrolmentProg	100%	1/1

***Tools and artifacts***



# Development environment: VS Code with GitHub SCM.



*View of project repo.*

The screenshot shows the GitHub interface for the repository 'kashmerchant/sprint'. The repository is marked as 'Private'. The navigation bar includes links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected, showing the file browser for the 'master' branch. The file browser displays a list of files and folders: '..' (root), 'aura' (11 minutes ago), 'classes' (4 minutes ago), 'lwc' (11 minutes ago), and 'triggers' (11 minutes ago). The 'classes' folder is highlighted, showing its contents: 'Typo' (4 minutes ago). The footer includes links for 'Terms', 'Privacy', 'Security', 'Status', 'Docs', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About', along with the GitHub logo and copyright notice '© 2022 GitHub, Inc.'.

Search or jump to...

Pulls Issues Codespaces Marketplace Explore

kashmerchant / sprint Private

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

master sprint / force-app / main / default / Go to file Add file ...

kashmerchant Typo ... 4 minutes ago History


..

aura	Questions 6, 7, 8, 9	11 minutes ago
classes	Typo	4 minutes ago
lwc	Questions 6, 7, 8, 9	11 minutes ago
triggers	Questions 6, 7, 8, 9	11 minutes ago




Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About


© 2022 GitHub, Inc.




*View of project repo.*













PullsIssuesCodespacesMarketplaceExplore




 kashmerchant / sprint Private


 Unwatch 1  Fork 0  Star 0

 Code  Issues  Pull requests  Actions  Projects  Security  Insights  Settings









 master  sprint / force-app / main / default / classes /

Go to file Add file ...


 kashmerchant Typo ...

5 minutes ago  History

..

 AddEnrolmentTest1.cls	Questions 6, 7, 8, 9	11 minutes ago
 AddEnrolmentTest1.cls-meta.xml	Questions 6, 7, 8, 9	11 minutes ago
 AddEnrolmentTest2.cls	Typo	5 minutes ago
 AddEnrolmentTest2.cls-meta.xml	Questions 6, 7, 8, 9	11 minutes ago
 StudentInitialisation.cls	Questions 6, 7, 8, 9	11 minutes ago
 StudentInitialisation.cls-meta.xml	Questions 6, 7, 8, 9	11 minutes ago
 StudentInitialisationTest.cls	Questions 6, 7, 8, 9	11 minutes ago
 StudentInitialisationTest.cls-meta.xml	Questions 6, 7, 8, 9	11 minutes ago

TermsPrivacySecurityStatusDocsContact GitHubPricingAPITrainingBlogAbout


 © 2022 GitHub, Inc.

***Artifact:** Doxygen-style documentation to clarify function specifications/interface.*

```
force-app > main > default > classes > StudentInitialisation.cls > ...
1  /**
2   * @author Kashmiri Merchant
3   * @description Methods for initialising Prior Qualification and Enrolment Progression fields (Question 7).
4   */
5  public class StudentInitialisation {
6
7      /**
8       * @description Randomly assigns a prior qualification to each record in studentList.
9       * @param studentList List of Student records.
10      */
11     public static void assignPriorQual(List<Contact> studentList) {
12
13         List<String> priorQuals = new List<String>{'Below Tertiary',
14                                                  'Tertiary Graduate',
15                                                  'Tertiary Post-Graduate'};
16
17         for (Contact student : studentList){
18
19             // Generating a random index into priorQuals list
20             Integer i = Integer.valueOf(Math.floor(Math.random()*priorQuals.size()));
21
22             student.Prior_Qualification__c = priorQuals[i];
23
24         }
25
26         update studentList;
27     }
28
29     /**
30     * @description Determines a student's enrolment progression. Called by the updateEnrolmentProg trigger.
31     * @param studentList List of Student records.
32     */
33     public static void updateEnrolmentProgHandler(List<Contact> studentList) {
34
35         for (Contact student : studentList){
36
37             switch on student.Prior_Qualification__c {      // Used switch in place of if-else for parsimony
38
39                 when 'Below Tertiary' {
40                     student.Enrolment_Progression__c = 'Advanced';
41                 }
42                 when 'Tertiary Graduate' {
43                     student.Enrolment_Progression__c = 'Ultimate';
44                 }
45                 when 'Tertiary Post-Graduate' {
46                     student.Enrolment_Progression__c = 'Extra';
47                 }
48                 when else {
49                     null;
50                 }
51             }
52
53         }
54
55     }
56
57 }
58
59 }
```

***Artifact:** View of  
generated  
documentation.*

Apex Documentation



Powered by [ApexDox VS Code](#)

Search...

[Home](#)

▼ Miscellaneous

[AddEnrolmentTest1](#)

[AddEnrolmentTest2](#)

**[StudentInitialisation](#)**

[StudentInitialisationTes](#)

Show: ☒ All ☒ global ☒ public ☒ protected ☒ private ☒ testMethod ☒ webService

## StudentInitialisation

Methods for intialising Prior Qualification and Enrolment Progression fields (Question 7).

**Signature**  
`public class StudentInitialisation`

**Author**  
Kashmira Merchant

### StudentInitialisation Methods

- [assignPriorQual\(studentList\)](#)  
Randomly assigns a prior qualification to each record in studentList.
- [updateEnrolmentProgHandler\(studentList\)](#)  
Determines a student's enrolment progression. Called by the updateEnrolmentProg trigger.

**assignPriorQual(studentList)**  
Randomly assigns a prior qualification to each record in studentList.

**Signature**  
`public static void assignPriorQual(List<Contact> studentList)`

**Parameters**  
`studentList`  
Type: `List<Contact>`  
List of Student records.

---

**updateEnrolmentProgHandler(studentList)**  
Determines a student's enrolment progression. Called by the updateEnrolmentProg trigger.

**Signature**  
`public static void updateEnrolmentProgHandler(List<Contact> studentList)`


**Parameters**  
`studentList`  
Type: `List<Contact>`  
List of Student records.

---

Powered By [ApexDox VS Code](#)

*Artifact: View of generated documentation.*

Apex Documentation



Powered by [ApexDox VS Code](#)

Search...

[Home](#)

▼ Miscellaneous

[AddEnrolmentTest1](#)

[AddEnrolmentTest2](#)

[StudentInitialisation](#)

[StudentInitialisationTest](#)

Show: ☒ All ☒ global ☒ public ☒ protected ☒ private ☒ testMethod ☒ webservice

## StudentInitialisationTest

Unit tests for StudentInitialisation class (Question 6)

**Signature**  
`@isTest`  
`private class StudentInitialisationTest`

**Author**  
Kashmira Merchant

### StudentInitialisationTest Methods

- `getStudent(studentId)`  
Helper method to get student whose record Id matches studentId.
- `testAdvanced()`  
Unit test 1 - assert('Tertiary Post-Graduate', 'Advanced')
- `testExtra()`  
Unit test 3 - assert('Below Tertiary', 'Extra')
- `testUltimate()`  
Unit test 2 - assert('Tertiary Graduate', 'Ultimate')

**getStudent(studentId)**  
Helper method to get student whose record Id matches studentId.

**Signature**  
`private static` Contact `getStudent(String studentId)`

**Parameters**  
`studentId`  
Type: `String`  
Record ID of the updated student.

---

**testAdvanced()**  
Unit test 1 - assert('Tertiary Post-Graduate', 'Advanced')

**Signature**  
`@isTest`  
`private static void testAdvanced()`

---


**testExtra()**  
Unit test 3 - assert('Below Tertiary', 'Extra')

**Signature**  
`@isTest`  
`private static void testExtra()`

*Artifact: Doxygen-style  
documentation*

*Function specifications.*

Apex Documentation



Powered by [ApexDox VS Code](#)

Q Search...

Home

▼ Miscellaneous

AddEnrolmentTest1

**AddEnrolmentTest2**

StudentInitialisation

StudentInitialisationTes

Show: ☒ All ☒ global ☒ public ☒ protected ☒ private ☒ testMethod ☒ webService

## AddEnrolmentTest2

Test for AddEnrolment trigger (v2).

**Signature**  
`@isTest`  
`private class AddEnrolmentTest2`

**Author**  
Kashmira Merchant

### AddEnrolmentTest2 Methods

- `testAddEnrolmentTrigger()`

**testAddEnrolmentTrigger()**

**Signature**  
`@isTest`  
`private static void testAddEnrolmentTrigger()`

---

[Powered By ApexDox VS Code](#)