

Unit 1: Review Basic Constructs in Apex

Learning Objectives

After completing this unit, you'll be able to:

- Use and apply Apex control flow statements.
- Describe the capabilities of the declarative process automation features

Key Topics

This module prepares you for the logic and process automation section of the Platform Developer I exam. The process automation and logic section, which makes up 38% of the overall exam. In this unit we will review these topics.

- Apex variables and constants
- Apex methods
- Interfaces


This module includes a number of interactive, real-world, scenario-based questions that are a lot like the ones you can encounter as a Salesforce developer. Looking at these scenarios helps prepare you to take the process automation and logic section of the Platform Developer I exam. As you tackle the practice questions, you get immediate feedback on your answers, along with detailed information on why your answers are correct (or incorrect).

This module includes three units on process automation and logic. This unit focuses on the basic constructs within Apex.

Download the Guide

Would you like a hard copy of the contents in these modules? Each module includes a link to a printable version you can download. Download the [Platform Developer I Certification Prep: Process Automation and Logic guide](#).

Exam Practice Questions

Ready to jump in? The sample tool below is not scored—it's just an easy way to quiz yourself. To use it, read the scenario and click the answer you think is correct. Some questions may have more than one correct answer. Click **Submit** and you get a popup telling you whether the answer you chose is correct or incorrect, and why; if there's a longer explanation, click  and then click anywhere in the window to close it. When you reach the end, you can review the answers or retake the questions.

Scenario 1

When the number of records in a recordset is unknown, which control statement should a developer use to implement a set of code that executes for every record in the recordset, without performing a .size() or .length() method call?

ANSWER	FEEDBACK
A. DO { ... } WHILE (CONDITION)	Incorrect. For do while, while, and traditional for loops, the developer must set the size value for the exit condition to eventually be met.
B. WHILE (CONDITION) {...}	Incorrect. For do while, while, and traditional for loops, the developer must set the size value for the exit condition to eventually be met.
C. FOR (INIT_STMT; EXIT_CONDITION; INCREMENT_STMT) { ... }	Incorrect. For do while, while, and traditional for loops, the developer must set the size value for the exit condition to eventually be met.
D. FOR (VARIABLE : LIST_OR_SET) { ... }	Correct. List or set iteration for loops do not need the size of the collection to run.

Scenario 2

What is the value of x after the code segment executes?

```
String x = 'A';

Integer i = 10;

if ( i < 15 ){

    i = 15;

    x = 'B';

} else

if ( i < 20 ){

    x = 'C';

} else {

    x = 'D';

}
```

ANSWER

FEEDBACK

A. 'A'	Incorrect. These if statements are evaluating the value of i. Based on the if statements the value of x will always be modified.
B. 'B'	Correct. These if statements are evaluating the value of i. Since i is equal to 10 then i is less than 15 and i will be reassigned the value 15 and x will be equal to 'B'. Once the values are assigned the if statement is no longer run.
C. 'C'	Incorrect. These if statements are evaluating the value of i. The value of x could be equal to 'C' because i is in fact less than 20. However, since this condition is in the else portion of the if else statement, the condition (i < 20) is never reached because the previous condition (i < 15) is met.
D. 'D'	Incorrect. These if statements are evaluating the value of i. The only way that x will be equal to 'D' is if all of the other conditions are false. Being that i is less than 15 and i is also less than 20 this else condition is never reached.

Scenario 3

Which three are accurate statements about variable scope? (Select three answers.)

ANSWER	FEEDBACK
A. A VARIABLE CAN BE DEFINED AT ANY POINT IN A BLOCK.	Correct. A variable is valid from the point where it is declared inside of the code.
B. SUB-BLOCKS CANNOT REUSE A PARENT BLOCK'S VARIABLE NAME.	Correct. Variables can be defined anywhere within a block. However Sub-blocks can't redefine a variable name that has already been used within a parent block.
C. SUB-BLOCKS CAN REUSE A PARENT BLOCK'S VARIABLE NAME IF ITS VALUE IS NULL.	Incorrect. Variables can be defined anywhere within a block. However Sub-blocks can't redefine a variable name that has already been used within a parent block regardless if the value is null.
D. PARALLEL BLOCKS CAN USE THE SAME VARIABLE NAME.	Correct. The same name can be used inside parallel blocks because the scope of the variable is limited only to the particular block that it was defined in.
E. A STATIC VARIABLE CAN RESTRICT THE SCOPE TO THE CURRENT BLOCK IF ITS VALUE IS NULL.	Incorrect. The value of a static variable persists within the context of a single transaction and is reset across transaction boundaries.

Exam Topic Flashcards

Use the interactive flashcards in each unit to brush up on some of the key topics you'll find on this portion of the exam.


Read the question or term on each card, then click the card to reveal the correct answer. Click the right-facing arrow to move to the next card, and the left-facing arrow to return to the previous card.

Question/Term	Answer/Definition
Which two users can edit a record after it has been locked for approval?	An administrator and a user who is assigned as the current approver
What happens to changes in the result when a Visualforce page calls an Apex controller, which calls another Apex class, and then hits a governor limit?	Any changes up to the error are rolled back.

Platform Developer I Certification Prep: Process Automation and Logic

Related Badges

Looking for more information? Explore these related badges.

Badge	Content Type
 Apex Basics & Database	Module

Next, let's review the second part of the process automation and logic section of the Platform Developer I exam.

Unit 2: Get a Refresh on SOQL, SOSL, and DML

Learning Objectives

After completing this unit, you'll be able to:

- Given a scenario, write SOSL, SOQL and DML statements in Apex.
- Identify the implications of governor limits on Apex transactions.
- Implement exception handling in Apex, including custom exceptions as needed.


Key Topics

This unit prepares you for these topics in the logic and process automation section of the Platform Developer I exam.

- SOQL
- SOSL
- DML
- Exceptions and governor limits

Exam Practice Questions

Ready to jump in? The sample tool below is not scored—it's just an easy way to quiz yourself. To use it, read the scenario and click the answer you think is correct. Some questions may have more than one correct answer. Click **Submit** and you get a popup telling you whether the answer you chose is correct or incorrect,

and why; if there's a longer explanation, click  and then click anywhere in the window to close it. When you reach the end, you can review the answers or retake the questions.

Scenario 1

A developer in a Salesforce org with 100 accounts executes the following code using the Developer Console.

```
Account myAccount = new Account(Name='MyAccount');
insert myAccount;

for (Integer x = 0; x < 150; x++){
    Account newAccount = new Account(Name='MyAccount' + x);
    try{
        insert newAccount;
    }catch (Exception ex){
        System.Debug(ex);
    }
}

insert new Account(Name='MyAccount');
```

How many accounts are in the org after this code is run?

ANSWER

FEEDBACK

A. 100	Correct. Salesforce has a governor limit of 150 DML statements per transaction. The Salesforce org originally starts with 100 accounts. The first DML statement inserts the first account leaving 149 DML statements available. Inside the for loop it tries to make 150 DML statements. But the limit is reached when x = 149. When this happens the transaction aborts and all data changes are rolled back. The only thing that is left is the original 100 accounts. Since the error happened the last insert statement is never reached. Best practice is to avoid DML statements inside for loops.
B. 101	Incorrect. Salesforce has governor limits of 150 DML statements per transaction. When the limit is reached, the transaction aborts and all changes are rolled back. Best practice is to avoid DML statements inside for loops.
C. 102	Incorrect. Salesforce has governor limits of 150 DML statements per transaction. When the limit is reached, the transaction aborts and all changes are rolled back. Best practice is to avoid DML statements inside for loops.
D. 252	Incorrect. Salesforce has governor limits of 150 DML statements per transaction. When the limit is reached, the transaction aborts and all changes are rolled back. Best practice is to avoid DML statements inside for loops.

Scenario 2

Which two statements should a developer avoid using inside procedural loops? (Select two answers.)

ANSWER	FEEDBACK
A. LIST<CONTACT> CONTACTS = [SELECT ID, SALUTATION, FIRSTNAME, LASTNAME, EMAIL FROM CONTACT WHERE ACCOUNTID = :A.ID];	Correct. Avoid using SOQL in loops since there is a governor limit that enforces a maximum number of SOQL queries. When these operations are placed inside a loop, database operations are invoked once per iteration of the loop making it very easy to reach these governor limits. Best practice is to query once to obtain all results, and then iterate over the results.
B. UPDATE CONTACTLIST;	Correct. Avoid using DML in loops since there is a governor limit that enforces a maximum number of DML statements. When these operations are placed inside a loop, database operations are invoked once per iteration of the loop making it very easy to reach these governor limits.
C. IF (O.ACCOUNTID == A.ID)	Incorrect. An if statement can be used inside of a loop to create a condition.
D. SYSTEM.DEBUG('AMOUNT OF CPU TIME (IN MS) USED SO FAR: ' + LIMITS.GETCPU TIME());	Incorrect. System.debug is a great way to test the output especially in loops.

Exam Topic Flashcards



The following flashcards cover SOQL, SOSL, exceptions, and governor limits. Use these interactive flashcards to review some of the key topics you'll find on this part of the exam.

Read the question or term on each card, then click on the card to reveal the correct answer. Click the right-facing arrow to move to the next card, and the left-facing arrow to return to the previous card.

Question/Term	Answer/Definition
Which data structure is returned to a developer when performing a SOSL search?	A list of lists of sObjects
A developer runs the following anonymous code block. <pre>List<Account> acc = [SELECT Id FROM Account LIMIT 10]; Delete acc; Database.emptyRecycleBin(acc); system.debug(Limits.getDMLStatements()+'', '+Limits.getLimitDMLStatements());</pre> What is the result?	2,150
A developer has the following query. <pre>Contact c = [SELECT id, firstname, lastname, email FROM Contact WHERE lastname = 'Smith'];</pre> What does the query return if there is no contact with the last name Smith?	An error that no rows are found.
A developer writes a Salesforce Object Query Language (SOQL) query to find child records for a specific parent. How many levels can be returned in a single query?	1

Related Badges

Looking for more information? Explore these related badges.

Badge	Content Type
 Database &.NET Basics	Module
 Search Solution Basics	Module

Great work! You're almost finished reviewing the process automation and logic section of the Platform Developer I exam. On to the last unit, which covers Apex classes and triggers.

Unit 3: Study Up on Apex Classes and Triggers

Learning Objectives

After completing this unit, you'll be able to:


- Write Apex classes and use Apex interfaces.
- Write Apex classes and triggers while following best practices.
- Describe the relationship between Apex transactions, the save order of execution, and the potential for recursion and cascading.

Key Topics

This unit prepares you for these topics in the logic and process automation section of the Platform Developer I exam.

- Apex classes
- Apex triggers
- Save order of execution

Exam Practice Questions

Ready to jump in? The sample tool below is not scored—it's just an easy way to quiz yourself. To use it, read the scenario and click the answer you think is correct. Some questions may have more than one correct answer. Click **Submit** and you get a popup telling you whether the answer you chose is correct or incorrect, and why; if there's a longer explanation, click  and then click anywhere in the window to close it. When you reach the end, you can review the answers or retake the questions.

Scenario 1

What is an accurate constructor for a custom controller named MyController?

ANSWER	FEEDBACK
A. <code>PUBLIC MYCONTROLLER(LIST<OBJECT> OBJECTS){ ACCOUNTS = (LIST<ACCOUNT>)OBJECTS;}</code>	Incorrect. This is the structure of a controller extension, not a custom controller.
B. <code>PUBLIC MYCONTROLLER(){ ACCOUNT = NEW ACCOUNT(); }</code>	Correct. To create a constructor for a custom controller, the constructor cannot have a parameter.
C. <code>PUBLIC MYCONTROLLER(SUBJECT OBJ){ ACCOUNT = (ACCOUNT) OBJ; }</code>	Incorrect. A custom controller constructor does not have a parameter.
D. <code>PUBLIC MYCONTROLLER(LIST<OBJECT> OBJECTS){ ACCOUNTS = (LIST<ACCOUNT>)OBJECTS; }</code>	Incorrect. A custom controller constructor does not have a parameter.

Scenario 2

A developer uses a before insert trigger on the Lead object to fetch the Territory__c object, where the Territory__c.PostalCode__c matches the Lead.PostalCode. The code fails when the developer uses the Apex Data Loader to insert 10,000 lead records. The developer has the following code block.

```
01 for(Lead l : Trigger.new){
02     if(l.PostalCode != null){
03         List<Territory__c> terrList = [SELECT Id FROM Territory__c WHERE
PostalCode__c = :l.PostalCode];
04         if(terrList.size() > 0){
05             l.Territory__c = terrList[0].Id;
06         }
07     }
08 }
```

Which line of code is causing the code block to fail?

ANSWER	FEEDBACK
A. 01: TRIGGER.NEW IS NOT VALID IN A BEFORE INSERT TRIGGER.	Incorrect. You can use trigger.new in a before insert trigger.
B. 02: A NULLPOINTER EXCEPTION IS THROWN IF POSTALCODE IS NULL.	Incorrect. This will not happen because in line 2, the code is making sure PostalCode is not null.
C. 03: A SOQL QUERY IS LOCATED INSIDE OF THE FOR LOOP CODE.	Correct. This will fail on the record number 201 because the limit of SOQL queries is 200. Best practice is to avoid SOQL queries inside for loops.
D. 05: THE LEAD IN A BEFORE INSERT TRIGGER CANNOT BE UPDATED.	Incorrect. The lead fields in a before insert trigger can be updated.

Scenario 3

A developer needs to automatically populate the ReportsTo field in a contact record based on the values of the related Account and Department fields in the contact record. Which two trigger types should the developer create? (Select two answers.)

ANSWER	FEEDBACK
A. BEFORE UPDATE	Correct. The developer should use a before update trigger. That way there is no need for an extra DML statement. It is good practice to use before triggers to validate updated records before they are saved.
B. BEFORE INSERT	Correct. The developer should use a before insert trigger. That way there is no need for an extra DML statement. It is good practice to use before triggers to validate records before they are created.
C. AFTER INSERT	Incorrect. After triggers should not be used to update fields on the record being created. After triggers are used to access field values that are set by the system and to affect changes in other records.
D. AFTER UPDATE	Incorrect. After triggers should not be used to update fields on the record being updated. After triggers are used to access field values that are set by the system and to affect changes in other records.

Scenario 4

A hierarchy custom setting stores a specific URL for each profile in Salesforce. Which statement can a developer use to retrieve the correct URL for the current user's profile and display this on a Visualforce Page?

ANSWER

FEEDBACK

A. `{!$SETUP.URL_SETTINGS__C.URL__C}`

Correct. A specific profile should get a specific record type. So, simply getting a record type Id will not work. We need to know what each profile should get. Hierarchy custom settings support unique values per profile.

B.
`{!$SETUP.URL_SETTINGS__C.INSTANCE[PROFILE.ID].URL__C}`

Incorrect. There is no need to specify the profile. Correct syntax is `{!$Setup.CustomSettingName__c.CustomFieldName__c}`.

C.
`{!$SETUP.URL_SETTINGS__C[PROFILE.ID].URL__C}`

Incorrect. There is no need to specify the profile. Correct syntax is `{!$Setup.CustomSettingName__c.CustomFieldName__c}`.

D.
`{!$SETUP.URL_SETTINGS__C[$PROFILE.ID].URL__C}`

Incorrect. There is no need to specify the profile. Correct syntax is `{!$Setup.CustomSettingName__c.CustomFieldName__c}`.

Scenario 5

In which order does Salesforce execute events upon saving a record?

ANSWER

A. VALIDATION RULES; BEFORE TRIGGERS; VALIDATION RULES; AFTER TRIGGERS; ASSIGNMENT RULES; WORKFLOW RULES; COMMIT

FEEDBACK

Correct.

Runs system validation check.

Executes all `before` triggers.

Runs most system validation steps again, such as verifying that all required fields have a non-null value, and runs any user-defined validation rules. The only system validation that Salesforce doesn't run a second time (when the request comes from a standard UI edit page) is the enforcement of layout-specific rules.

Executes duplicate rules. If the duplicate rule identifies the record as a duplicate and uses the block action, the record is not saved and no further steps, such as `after` triggers and workflow rules, are taken.

Saves the record to the database, but doesn't commit yet.

Executes all `after` triggers.

Executes assignment rules.

Executes auto-response rules.

Executes workflow rules.

If there are workflow field updates, updates the record again.

If the record was updated with workflow field updates, fires `before` update triggers and `after` update triggers one more time (and only one more time), in addition to standard validations. Custom validation rules, duplicate rules, and escalation rules are not run again.

Platform Developer I Certification Prep: Process Automation and Logic

B. BEFORE TRIGGERS; VALIDATION RULES; AFTER TRIGGERS; WORKFLOW RULES; ASSIGNMENT RULES; COMMIT

Incorrect. Once the after triggers are executed, the assignment rules execute.

C. VALIDATION RULES; BEFORE TRIGGERS; AFTER TRIGGERS; WORKFLOW RULES; ASSIGNMENT RULES; COMMIT

Incorrect. Once the after triggers are executed, the assignment rules execute.

D. BEFORE TRIGGERS; VALIDATION RULES; ASSIGNMENT RULES; WORKFLOW RULES; AFTER TRIGGERS; COMMIT

Incorrect. There are validation rules that happen before the before triggers execute that are system level validations, this includes page level validations and field length checks. Once the after triggers are executed, the assignment rules execute.

Scenario 6

How can a developer determine if a CustomObject__c record has been manually shared with the current user in Apex?

ANSWER	FEEDBACK
A. BY CALLING THE PROFILE SETTINGS OF THE CURRENT USER	Incorrect. Profile does not provide record sharing information.
B. BY QUERYING CUSTOMOBJECT__SHARE	Correct. To access sharing programmatically, you must use the share object associated with the standard or custom object for which you want to share. All custom object sharing objects are named MyCustomObject__Share, where MyCustomObject is the name of the custom object.
C. BY CALLING THE ISSHARED() METHOD FOR THE RECORD	Incorrect. The isShared method does not exist for this functionality.
D. BY QUERYING THE ROLE HIERARCHY	Incorrect. Role hierarchy alone does not give enough information on sharing.

Scenario 7

A developer creates a workflow rule declaratively that changes the value of a field on an object. An Apex after update trigger exists for the object. What happens when a user updates a record?

ANSWER	FEEDBACK
A. THE APEX TRIGGER IS FIRED MORE THAN ONCE.	Correct. After the workflow rule, the trigger is executed again.
B. THE WORKFLOW RULE IS FIRED MORE THAN ONCE.	Incorrect. The workflow rule is only executed once.
C. NO CHANGES ARE MADE TO THE DATA.	Incorrect. Changes are made, since the workflow rule is executed and then the trigger.
D. BOTH THE APEX TRIGGER AND WORKFLOW RULE ARE FIRED ONLY ONCE.	Incorrect. The workflow rule is fired once, but the trigger is executed twice.

Scenario 8

A developer wants to display all of the available record types for a Case object. The developer also wants to display the picklist values for the Case.Status field. The Case object and the Case.Status field are on a custom Visualforce page.

Which two actions should the developer perform to get the record types and picklist values in the controller? (Select two answers.)

ANSWER	FEEDBACK
A. USE SCHEMA.RECORDTYPEINFO RETURNED BY CASE.SOBJECTTYPE.GETDESCRIBE().GETRECORDTYPEINFOS().	Correct. A RecordTypeInfo object is returned from the sObject describe result using the getRecordTypeInfos method.
B. USE SOQL TO QUERY CASE RECORDS IN THE ORG TO GET ALL THE RECORDTYPE VALUES AVAILABLE FOR CASE.	Incorrect. This is unnecessary and very inefficient since this information can be gathered using Case.SObjectType.getDescribe().getRecordTypeInfos().
C. USE SCHEMA.PICKLISTENTRY RETURNED BY CASE.STATUS.GETDESCRIBE().GETPICKLISTVALUES().	Correct. getPicklistValues() returns a list of PicklistEntry objects.
D. USE SOQL TO QUERY CASE RECORDS IN THE ORG TO GET ALL VALUES FOR THE STATUS PICKLIST FIELD.	Incorrect. This is unnecessary and very inefficient since this information can be gathered using Case.Status.getDescribe().getPicklistValues().

Exam Topic Flashcards

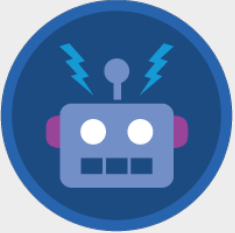


The flashcards in this unit cover Apex classes and triggers. Use these interactive flashcards to brush up on some of the key topics you'll find on this part of the exam.

Read the question or term on each card, then click the card to reveal the correct answer. Click the right-facing arrow to move to the next card, and the left-facing arrow to return to the previous card.

Question/Term	Answer/Definition
A developer writes a before insert trigger. Which context variable can the developer use to access the incoming records in the trigger body?	The <code>Trigger.new</code> context variable
Which method can a developer use to determine, from the <code>DescribeSObjectResult</code>, if the current user will be able to create records for an object in Apex?	The <code>isCreatable()</code> method
In the execution order of triggers, what three steps happen after the <code>before</code> triggers execute, and before the <code>after</code> triggers execute?	<ol style="list-style-type: none">1. <code>Before</code> triggers are executed.2. System validation steps are run again and user-defined validation rules are checked.3. Executes duplicate rules.4. The record is saved to the database but doesn't commit yet.5. Executes all <code>after</code> triggers.
True or False: Workflows support publishing platform events.	False

Related Badges

Looking for more information? Explore these related badges.

Badge	Content Type
 Lightning Flow	Module
 Apex Triggers	Module
 Apex Specialist	Superbadge

Congratulations! You've studied up on the process automation and logic section of the Platform Developer I certification exam.