

/ Salesforce / By SkillCertPro

## Practice Set 6

Your results are here!! for " Salesforce Platform Developer 1 Practice Test 6 "

0 of 58 questions answered correctly

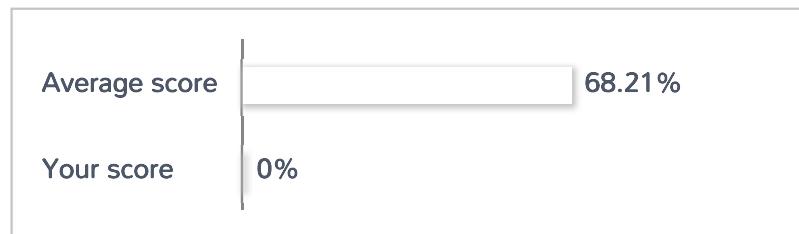
Your time: 00:00:14

Your Final Score is : 0

You have attempted : 0

Number of Correct Questions : 0 and scored 0

Number of Incorrect Questions : 0 and Negative marks 0



You can review your answers by clicking view questions.

**Important Note :** Open Reference Documentation Links in New Tab (Right Click and Open in New Tab).

[Restart Test](#)

[View Answers](#)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34

Answered      Review

[Review question](#)

[Pause](#)

[Summary](#)



## 1. Question

Which of the following is a PRIMITIVE data type?

- A. Enum
- B. Map
- C. List
- D. Set
- E. Long

**Unattempted**

**Correct Answer:** E. Long

Long is a primitive data type in Apex, representing a 64-bit integer. Primitive data types are fundamental building blocks in Apex and are directly manipulated. Knowing these is crucial for the Platform Developer I exam.

**Incorrect Options:**

A. Enum

Enums are a way to define a set of named constants. While useful, they are *not* primitive data types. They are a type of custom type built upon other types. The Platform Developer I exam often tests understanding of the difference between primitive and other data types.

B. Map

A Map is a collection of key-value pairs. It's a collection type, specifically used for storing and retrieving data based on keys. Collection types like Map, List, and Set are not primitive data types. Understanding the distinction is key for the exam.

C. List

A List is an ordered collection of elements. It's a collection type, used for storing multiple values in a specific sequence. Again, collection types are not primitive. The Platform Developer I exam expects you to know this.

D. Set

A Set is an unordered collection of unique elements. It's a collection type, useful for ensuring that only distinct values are stored. Like other collection types, Set is not a primitive data type. This is a common area

tested on the exam.

## 2. Question

Which can be invoked via a Workflow Rule?

- Apex
- Neither
- Process Builder
- Flow

Unattempted

Correct:

**B. Neither**

Workflow Rules cannot invoke **Apex**, **Process Builder**, or **Flow** directly. They are limited to their own set of actions, such as sending email alerts, updating fields, or creating tasks. Workflow Rules are designed for simpler automation and do not have the capability to trigger more complex tools like Apex, Process Builder, or Flows.

Incorrect:

**X A. Apex**

Workflow Rules cannot invoke Apex code. Apex can only be triggered by other mechanisms, such as triggers or Process Builder, but not directly by Workflow Rules.

**X C. Process Builder**

Workflow Rules cannot invoke Process Builder. Process Builder is a more advanced automation tool that can trigger Flows, Apex, and other processes, but it is not something that can be invoked by a Workflow Rule.

**X D. Flow**

Workflow Rules cannot invoke Flows directly. Flows can be triggered by Process Builder, record changes, or other events, but not by Workflow Rules.

## 3. Question

Which method can be used to retrieve parameters passed to the URL from the controller to the Visualforce page?

- .getpoint
- .getanchor

- .getreference
- .getParameters

**Unattempted**

**Correct Answer:**

- D. .getParameters

The `.getParameters()` method is the correct way to retrieve URL parameters in a Visualforce controller. This method returns a `Map<String, String>` where the keys are the parameter names and the values are the parameter values. This is essential knowledge for the Platform Developer I exam.

**Incorrect Options:**

- A. .getpoint

`.getpoint` is not a valid method for retrieving URL parameters. It's not related to URL parameter handling. The Platform Developer I exam will often include misleading options like this.

- B. .getanchor

`.getanchor` might seem related to URLs, but it's used for handling URL fragments (the part after the `#` symbol). It's not used for retrieving query parameters. Distinguishing between fragments and parameters is important for the exam.

- C. .getreference

`.getreference` is used for generating URLs within Salesforce, often for linking to other Visualforce pages or Salesforce resources. It's not used for *retrieving* parameters. The Platform Developer I exam often tests your knowledge of various URL-related methods, so understanding their specific purposes is crucial.

## 4. Question

Can custom objects, fields and relationships be created directly from the Schema Builder?

- No
- Yes

**Unattempted**

- Yes

The Schema Builder is a visual tool within Salesforce that allows you to create and manage custom objects, fields, and relationships directly. This is a fundamental aspect of Salesforce development and is definitely

covered on the Platform Developer I exam.

## 5. Question

What is the storage limit of a Developer sandbox?

- 100MB
- 300MB
- 200MB
- 200GB

Unattempted

Correct:

C. 200MB

A **Developer Sandbox** in Salesforce has a **storage limit of 200MB**. This includes **50MB of data storage** and **150MB of file storage**. Developer Sandboxes are mainly used for coding and testing in an isolated environment without affecting production data.

Incorrect:

A. 100MB – This is incorrect because the actual storage limit of a Developer Sandbox is **200MB**, not 100MB.

B. 300MB – This is incorrect because a Developer Sandbox has **200MB** of total storage, not 300MB.

D. 200GB – This is incorrect as **200GB is significantly larger than the actual limit**. No sandbox type has this limit; even **Full Copy Sandboxes** have limits based on production storage.

## 6. Question

Up to how many contacts can be merged at one time in Salesforce?

- 5
- 2
- 10
- 3

Unattempted

Correct Answer:  D. 3

You can merge up to three contacts at a time in Salesforce. This is a specific limit and a common topic on the Platform Developer I exam.

**Incorrect Options:**

- ✗ A. 5 ✗ B. 2 ✗ C. 10

These options are incorrect. While other Salesforce features might have different limits, the contact merge limit is specifically three. The Platform Developer I exam often tests knowledge of these specific limits, so memorizing them is important.

## 7. Question

Which one is NOT one of the three types of trace flags that can be used by a developer?

- Class Based
- User Based
- Test Based
- Trigger Based

**Unattempted****Correct Answer:**

- C. Test Based

Trace flags are primarily User-Based, Apex Class-Based, and Trigger-Based. “Test Based” is not a recognized type of trace flag in Salesforce. This is a common question on the Platform Developer I exam, testing your understanding of debugging and trace flags.

**Incorrect Options:**

- ✗ A. Class Based

Class-based trace flags allow you to set debug logging specifically for a particular Apex class. This is a valid and frequently used trace flag type.

- ✗ B. User Based

User-based trace flags allow you to set debug logging for a specific user, capturing all their activity within Salesforce. This is essential for debugging user-specific issues.

- ✗ D. Trigger Based

Trigger-based trace flags allow you to set debug logging specifically for Apex triggers. This is crucial for debugging trigger logic.

## 8. Question

Which access modifier allows declaring with a signature and no body?

- Both
- Abstract
- Neither
- Virtual

**Unattempted**

**Correct Answer:**

- B. Abstract

The **abstract** keyword allows you to declare a method signature without providing a body. Abstract methods *must* be implemented by a concrete (non-abstract) subclass. This is a core concept for the Platform Developer I exam, as it relates to object-oriented programming principles in Apex.

**Incorrect Options:**

- A. Both

While **virtual** and **abstract** are related, **virtual** alone does *not* allow you to declare a method without a body. **virtual** methods can have a body, and they can be overridden in subclasses.

- C. Neither

This is incorrect, as **abstract** specifically allows for method declarations without a body.

- D. Virtual

The **virtual** keyword allows a method to be overridden in a subclass, but it doesn't require the method to be declared without a body. A **virtual** method *can* have a body. The distinction between **virtual** and **abstract** is frequently tested on the Platform Developer I exam.

## 9. Question

The Junction Object used between the two objects joined by a many-to-many relationship will have what type of relationship between both those objects?

A. Master Detail B. Many to Many C. Lookup D. Hierarchical**Unattempted****Correct Answer:** A. Master Detail

The junction object in a many-to-many relationship has a *Master-Detail* relationship with *both* of the objects it connects. This is fundamental to how many-to-many relationships are implemented in Salesforce and is a very common topic on the Platform Developer I exam.

**Incorrect Options:** B. Many to Many

The junction object itself does *not* have a many-to-many relationship. It's the *combination* of the two Master-Detail relationships on the junction object that creates the many-to-many relationship between the original two objects. This is a critical distinction to understand for the exam.

 C. Lookup

While you *could* technically use Lookup relationships, Master-Detail relationships are *required* for properly implementing a many-to-many relationship in Salesforce. Master-Detail relationships enforce referential integrity and control deletion behavior, which are essential for this type of relationship. The Platform Developer I exam will almost certainly test this.

 D. Hierarchical

Hierarchical relationships are used for special cases like managing relationships between users in a role hierarchy. They are not relevant for implementing many-to-many relationships. Confusing these is a common mistake that the exam might try to exploit.

## 10. Question

Which of the following Visualforce page components can be used to display a standard record detail page of a particular object?

 A. apex:detail

- apex:single
- apex:simple
- apex:record

**Unattempted****Correct Answer:**

- A. apex:detail

The `<apex:detail>` component is specifically designed to display the standard detail page layout for a record in Visualforce. This is a key component for the Platform Developer I exam.

**Incorrect Options:**

- B. apex:single

There's no Visualforce component called `<apex:single>`. The Platform Developer I exam often includes non-existent components as distractors.

- C. apex:simple

There's no Visualforce component called `<apex:simple>`. Again, this is a distractor.

- D. apex:record

The `<apex:record>` component is used for displaying and editing records, but it provides a more customizable interface. While you *could* use it to display record details, it's not the component specifically intended for replicating the standard detail page layout. `<apex:detail>` is the most direct and appropriate component for this purpose, and the exam will likely expect you to know this.

## 11. Question

What are Connectors in the Flow Builder?

- They Define the behaviour of the flow, they are the building blocks for defining various actions.
- Are the paths defined between the various elements that represent the order of execution.
- It is the start element which can in some flows be programmed to regulate when the Flow should trigger.
- Are containers that hold data values. Like variables.

**Unattempted****Correct Answer:**

- B. Are the paths defined between the various elements that represent the order of execution.

Connectors in Flow Builder visually represent the flow of logic. They determine the order in which elements (like actions, decisions, and loops) are executed. This is a fundamental concept for the Platform Developer I exam.

**Incorrect Options:**

- A. They Define the behavior of the flow, they are the building blocks for defining various actions.

This describes the *elements* within a flow (like actions and decisions), not the connectors. Connectors define the *relationship* between elements, not the elements themselves. This distinction is important for the exam.

- C. It is the start element which can in some flows be programmed to regulate when the Flow should trigger.

This describes the *Start* element of a flow, not connectors. The Start element initiates the flow, but connectors determine the subsequent path. Confusing these is a common mistake.

- D. Are containers that hold data values. Like variables.

This describes *variables* within a flow, not connectors. Variables store data used by the flow's elements. Connectors, on the other hand, define the flow's path. The Platform Developer I exam often tests your understanding of these different flow components.

## 12. Question

Which Strategy Builder Action is defined here: “used to modify recommendations on the fly, when a strategy is executed.”

- Generate
- Load
- Branch Merge
- Enhance
- Branch Selector

**Unattempted**

**Correct Answer:**

- D. Enhance

The “Enhance” action in Strategy Builder is specifically designed to modify recommendations dynamically during strategy execution. This is a key concept for understanding how strategies work and is relevant for the

Platform Developer I exam.

#### Incorrect Options:

A. Generate

The “Generate” action is used to *create* recommendations, not modify existing ones. It’s a foundational action but doesn’t fit the definition given in the question.

B. Load

The “Load” action retrieves data or recommendations from a source, but it doesn’t directly modify them. It’s used for bringing data *into* the strategy, not changing it.

C. Branch Merge

The “Branch Merge” action combines the results of different branches within a strategy. It’s about combining paths, not modifying recommendations themselves.

E. Branch Selector

The “Branch Selector” action chooses which branch of a strategy to follow based on certain conditions. It controls the flow of execution, not the modification of recommendations. The Platform Developer I exam often includes questions about the specific purposes of different Strategy Builder actions, so understanding these distinctions is important.

### 13. Question

What does it mean that in interfaces the methods only contain the signature?

- It means that the signature of the method is empty, something that CANNOT be done with Apex Classes
- It means that the body of the method is empty, something that CAN be done with Apex Classes
- It means that the body of the method is full, something that CANNOT be done with Apex Classes
- It means that the body of the method is empty, something that CANNOT be done with Apex Classes

#### Unattempted

#### Correct Answer:

- D. It means that the body of the method is empty, something that CANNOT be done with Apex Classes

Interfaces in Apex (and many other object-oriented languages) define a contract. They specify *what* a class implementing the interface must do (the method signatures), but not *how* it should do it. The “how” (the

method body) is provided by the classes that implement the interface. This is a fundamental concept for the Platform Developer I exam.

**Incorrect Options:**

✗ A. It means that the signature of the method is empty, something that CANNOT be done with Apex Classes

An empty signature wouldn't define a method at all. Interfaces must have method signatures (name, parameters, return type).

✗ B. It means that the body of the method is empty, something that CAN be done with Apex Classes

While Apex *classes* can have methods with empty bodies (abstract methods), this is not the defining characteristic of *interfaces*. Interfaces *must* have empty method bodies. This subtle distinction is important for the exam.

✗ C. It means that the body of the method is full, something that CANNOT be done with Apex Classes

A "full" method body means the method has implementation. Interfaces, by definition, do *not* have implementation. The exam often tests this core understanding of interfaces.

## 14. Question

What is required to allow users to interact and view the Einstein Next Best Action element?

- A Visualforce Page
- An Aura Component
- A Lightning Component
- An Apex Class

**Unattempted**

**Correct:**

C. A Lightning Component.

Lightning Components (both LWC and Aura, though LWC is preferred) are the standard way to build user interfaces within the Lightning Experience and Salesforce mobile app. Einstein Next Best Action (NBA) recommendations are displayed and interacted with through components. You'd use the `lightning-flow` component (or its predecessor `force:flow`) within your custom Lightning Web Component (LWC) or Aura component to embed the NBA functionality. This is the primary way a user interacts with NBA recommendations.

Incorrect:

**X A. A Visualforce Page.**

Visualforce Pages are an older technology primarily used for classic Salesforce interfaces. While technically you *might* be able to embed a Lightning component (which would then handle the NBA display) within a Visualforce page, it's not the recommended or modern approach for working with NBA. It adds complexity and is generally avoided in modern Salesforce development. The exam will likely be looking for the most current best practice.

**X B. An Aura Component.**

Aura Components are a predecessor to Lightning Web Components. While NBA *can* be used with Aura components, Lightning Web Components are now the standard and preferred way to develop Lightning Experience interfaces. So, while not strictly *incorrect* in that it *could* work, it's not the *best* answer and generally wouldn't be the answer Salesforce is looking for on a modern exam focused on current best practices. LWC would be the preferred answer.

**X D. An Apex Class.**

Apex is server-side code. While Apex *can* be used to *retrieve* NBA recommendations, it's not directly involved in *displaying* them to the user. Apex would be used behind the scenes by a Lightning component to fetch the recommendations, but the component itself is what the user interacts with. Apex handles the *data*, the component handles the *display*.

## 15. Question

Following the best practices for iterating through collections can elements of a collection be modified as the loop iterates through them?

No

Yes

**Unattempted**

No

Modifying elements of a collection while iterating through them can lead to runtime exceptions, such as `ConcurrentModificationException`, or unintended behavior. In Salesforce Apex, when iterating through collections like Lists or Sets using `for-each` loops, modifying the collection (e.g., adding or removing elements) during iteration is not allowed. Instead, developers should use methods like creating a temporary list to store elements that need modification and updating the collection after the loop completes.

## 16. Question

Which of the following cannot be uploaded as a static resource and referenced in a Visualforce page using a global variable?

- Archive
- Style sheet
- javascript file
- apex class

Unattempted

Correct:

D. Apex Class

Apex classes are server-side code and cannot be stored or accessed as static resources. Static resources are for storing static content like images, stylesheets, JavaScript files, and other non-executable files that are used on the client-side (Visualforce pages, Lightning components, etc.). Apex classes are compiled and stored separately on the Salesforce platform.

Incorrect:

A. Archive

Archives (like .zip files) *can* be uploaded as static resources. This is commonly used for bundling multiple files (like images, CSS, and JS) into a single resource for easier management and deployment.

B. Style sheet

CSS stylesheets are a common use case for static resources. They can be uploaded and then referenced in Visualforce pages to style the page's appearance.

C. JavaScript file

JavaScript files are also frequently stored as static resources. This allows you to include JavaScript libraries or custom scripts in your Visualforce pages.

## 17. Question

Can Processes from Process Builder be migrated in active mode to new org?

- No
- Yes

**Unattempted**

A. No

Processes (including those built with Process Builder) cannot be migrated in *active* mode to a new org. When migrating processes (or flows, which have largely replaced Process Builder) using tools like change sets or the Metadata API, they are deployed in an *inactive* state in the target org. This prevents them from immediately firing and potentially causing issues in the new environment before they've been reviewed and tested.

**18. Question**

Which Object in Salesforce combines certain fields of Contacts and Accounts into a single object used for B2C needs?

- Business Accounts
- Contacts
- Social Accounts
- Person Accounts

**Unattempted**

Correct:

D. Person Accounts

Person Accounts are specifically designed for B2C scenarios where you're dealing with individual consumers rather than businesses. They combine fields from both Accounts and Contacts into a single record, simplifying data management when each customer is primarily an individual. They effectively treat a person as both an Account (for billing, etc.) and a Contact (for communication).

Incorrect:

A. Business Accounts

“Business Accounts” is not a standard Salesforce object. The standard object for representing businesses is simply “Account.” While you might customize Account record types or page layouts to tailor them for different business types, there’s no object called “Business Account.”

B. Contacts

Contacts represent individuals associated with an Account (business). While crucial for B2B and even relevant in B2C, they don’t combine the Account information in the way Person Accounts do. They are *related* to Accounts, but distinct from them.

### ✗ C. Social Accounts

“Social Accounts” isn’t a standard Salesforce object. Salesforce has features for integrating with social media, but they don’t involve an object called “Social Account” that merges Account and Contact fields. Social information is typically linked to standard Accounts or Contacts.

## 19. Question

Which loop would you use if you wanted to execute a code block as long as condition is true?

- A List/Set Iteration Loop
- B SOQL Loop
- C While Loop
- D Traditional Loop
- E Do-While Loop

Unattempted

Correct:

✓ C. While Loop

A while loop is designed to execute a block of code as long as a specified condition is true. The condition is evaluated before the loop body is executed, and if the condition is false initially, the loop body will not execute at all. This makes it the ideal choice when you want to execute code based on a condition being true.

Incorrect:

✗ A. List/Set Iteration Loop

This type of loop is used to iterate over elements in a list or set. It does not rely on a condition to continue executing; instead, it iterates over all elements in the collection. It is not suitable for executing code based on a condition being true.

✗ B. SOQL Loop

A SOQL loop is used to iterate over records returned by a SOQL query. It is not controlled by a condition but rather by the number of records returned by the query. It is not designed to execute code based on a condition being true.

✗ D. Traditional Loop

A “traditional loop” typically refers to a `for` loop, which executes a fixed number of times based on an initialization, condition, and increment/decrement. It is not ideal for executing code solely based on a condition being true, as it also depends on the initialization and increment/decrement.

### ✗ E. Do-While Loop

A do-while loop is similar to a while loop but guarantees that the loop body will execute at least once because the condition is evaluated after the loop body. While it does execute based on a condition, it is not the best answer because the question does not specify that the loop body must execute at least once.

## 20. Question

What Formula Field Function can be used to include conditional logic onto a field?

- IF
- ELSE
- SWITCH
- BUT

**Unattempted**

**Correct:**

- A. IF

The IF function is the core function for conditional logic in Salesforce formula fields. It allows you to define a condition and specify different values or calculations based on whether the condition is true or false.

**Incorrect:**

- B. ELSE

ELSE is not a standalone formula function. It's a *part* of the IF function's syntax. You use ELSE to specify the result when the IF condition is false, but you can't use ELSE on its own. It's always paired with an IF.

- C. SWITCH

SWITCH is used for more complex conditional logic where you have multiple possible values for a field and want to return different results for each value. While it provides conditional logic, the question asked for a function that can *include* conditional logic. The IF function is more fundamental and can be used within a SWITCH statement. SWITCH is not as generally applicable as IF for simple conditional logic.

- D. BUT

BUT is not a Salesforce formula function at all. It's a conjunction in English, but it has no relevance in the context of formula fields.

## 21. Question

Which tab in the Execution Overview log panel provides several metrics including the execution time (total, avg, min, max) for methods?

- Source
- Execution Units
- Execution Stack

Unattempted

Correct:

B. Execution Units

The Execution Units tab in the debug log provides detailed performance metrics for various parts of your code execution, including the execution time (total, average, minimum, and maximum) for methods, triggers, and other executable units. This is the tab you'd use to analyze performance bottlenecks and optimize your code.

Incorrect:

A. Source

The Source tab displays the actual code that was executed. While helpful for understanding the flow of execution, it doesn't directly provide the aggregate performance metrics (total, average, min, max times) for methods. You'd see the code, but not the summarized timing information.

C. Execution Stack

The Execution Stack tab shows the order in which methods and triggers were called (the call stack). This is useful for understanding the sequence of events, but it doesn't provide the detailed execution time metrics for individual methods that the Execution Units tab offers. It gives you the *call tree*, but not the *timing data* for the calls.

## 22. Question

Are unit tests subject to governor limits?

- Yes
- No

Unattempted

Yes

Unit tests are subject to governor limits in Salesforce. While some limits are more relaxed for testing (e.g., SOQL query limits), unit tests still operate within the boundaries of governor limits. This is crucial because it ensures that your tests realistically simulate the behavior of your code in a production environment and helps prevent code that might exceed limits when deployed.

## 23. Question

Which of the following is NOT a standard relationship type?

- A. Lookup
- B. Indirect Lookup
- C. Master Detail
- D. Master Lookup

Unattempted

Correct:

 D. Master Lookup

“Master Lookup” is not a standard Salesforce relationship type. There are Lookup relationships and Master-Detail relationships, but not a combination with that specific name.

Incorrect:

 A. Lookup

Lookup relationships are a standard relationship type in Salesforce. They allow you to link two objects together, but without the strong parent-child dependency of a Master-Detail relationship. They are a key part of the platform.

 B. Indirect Lookup

Indirect Lookup relationships are also a standard (though more advanced) relationship type. They allow you to connect objects through a lookup on a related object, creating a chain of relationships. They do exist and are used in specific scenarios.

 C. Master Detail

Master-Detail relationships are a core standard relationship type. They create a strong parent-child link between two objects, where the child record is dependent on the parent (e.g., deleting the parent deletes the child).

## 24. Question

Which Strategy Builder Action is defined here: “can load an existing recommendation, already pre-created, generally use after a Branch Selector element, which creates a division selection.”

- Generate
- Enhance
- Load
- Branch Selector
- Branch Merge

**Unattempted**

**Correct:**

**C. Load**

The description “can load an existing recommendation, already pre-created, generally use after a Branch Selector element, which creates a division selection” perfectly defines the **Load** strategy builder action. It’s used to retrieve and display a pre-existing, configured recommendation.

**Incorrect:**

**X A. Generate**

The Generate action is used to *create* a new recommendation based on defined criteria. It doesn’t load an existing one.

**X B. Enhance**

The Enhance action is typically used to *modify* or enrich an existing recommendation. While it works with existing recommendations, it’s not primarily for *loading* them. Its purpose is to add or change details.

**X D. Branch Selector**

The Branch Selector is a *decision element* in Strategy Builder, not an action. It directs the flow of the strategy based on conditions, but it doesn’t load recommendations itself. The Load action often follows a Branch Selector to handle different paths.

**X E. Branch Merge**

The Branch Merge element is used to *combine* different branches of a strategy back together. Like the Branch Selector, it’s a structural element, not an action that loads recommendations. It’s about the flow of

the strategy, not the retrieval of recommendations.

## 25. Question

Can the Process Builder have scheduled actions?

Yes

No

**Unattempted**

Yes

Process Builder *can* have scheduled actions. You can configure a process to run at a specific time or on a recurring schedule. This allows you to automate tasks like sending reminders, updating records, or performing other actions based on a schedule.

## 26. Question

Can Custom Buttons be used with standard AND custom objects?

No

Yes

**Unattempted**

Yes

Custom buttons can be created and used with both standard and custom objects in Salesforce. This is a fundamental part of customizing the user interface and adding specific functionality to object pages.

## 27. Question

By default the Process Builder will be imported to another org in which state?

Inactive

Active

**Unattempted**

A. Inactive

Processes (including those built with Process Builder, and now Flows) are *always* deployed or migrated to another org in an *inactive* state by default. This is a crucial safety measure.

## 28. Question

What is required of the standard controller of the page to allow a Visualforce page to be embedded on the Page layout?

- The controller of the Visualforce page must match the object.
- All other classes in the Visualforce page must match the same class as in the controller.

**Unattempted**

**Correct:**

- A. The controller of the Visualforce page must match the object.

To embed a Visualforce page on a page layout, the Visualforce page's standard controller (or a custom controller that extends the standard controller) *must* be associated with the same object as the page layout. This is how Salesforce knows which record's context to use when rendering the embedded page.

**Incorrect:**

- B. All other classes in the Visualforce page must match the same class as in the controller.

This is completely incorrect. There's no such requirement. A Visualforce page can use any Apex classes necessary for its functionality, regardless of what the main controller is. The controller simply provides the initial data context (the record being viewed on the page layout). Other classes can be used for various purposes (e.g., helper classes, classes for calling web services, etc.) and are not constrained in this way. They are independent of the controller in terms of this embedding requirement.

## 29. Question

Do Visualforce Pages support Delete actions?

- Yes
- No

**Unattempted**

**Yes**, Visualforce Pages support Delete actions. You can use the `apex:commandButton` component with the `action="delete"` attribute to trigger a delete operation on a record.

Here's a basic example:

**HTML**

```
<apex:page standardController="Account">
    <apex:form>
        <apex:commandButton action="{!!delete}" value="Delete"/>
    </apex:form>
</apex:page>
```

**Use code with caution.**

This code will display a button labeled “Delete” on the page. When clicked, it will delete the current Account record.

**Note:** To ensure data integrity and prevent accidental deletions, it's often recommended to implement confirmation dialogs or other safeguards before executing delete operations.

### 30. Question

For which STANDARD objects relationships are Rollup-Summary Fields NOT available?

- A. Account and Case
- B. Campaigns and Campaign Member
- C. Accounts and Opps
- D. Opp and Product

Unattempted

Correct:

B. Campaigns and Campaign Member

Roll-up summary fields are *not* available on the relationship between Campaigns and Campaign Members.

This is a specific limitation in Salesforce.

Incorrect:

A. Account and Case

Roll-up summary fields *are* available on the standard relationship between Accounts and Cases. You can roll up information from related Cases to the Account.

C. Accounts and Opps

Roll-up summary fields *are also* available on the standard relationship between Accounts and Opportunities. This is a very common use case (e.g., rolling up the total amount of related Opportunities to the Account).

**X D. Opp and Product** (Technically, this should be *Opportunity and Opportunity Line Item*)

While the question uses “Product,” it refers to the relationship between Opportunities and Opportunity Line Items (formerly called Opportunity Products). Roll-up summary fields are available on this relationship. You can roll up quantities, amounts, etc., from the Opportunity Line Items to the Opportunity.

### 31. Question

What can be incorporated into an apex class to make it so that not all records are shown but only those that the user has access to via the sharing rules?

- with sharing
- without sharing
- multiple sharing
- inherited sharing

**Unattempted**

**Correct:**

- A. with sharing

The `with sharing` keyword is used in Apex classes to enforce sharing rules. When a class is declared `with sharing`, any SOQL queries performed within that class will respect the user's sharing rules, meaning only records the user has access to will be returned.

**Incorrect:**

- B. without sharing

The `without sharing` keyword explicitly disables sharing rules. Code in a `without sharing` class will run in system context and ignore the user's sharing rules, potentially accessing data the user shouldn't see. This is generally used for utility classes or specific scenarios where bypassing sharing is necessary, but it should be used with extreme caution.

- C. multiple sharing

`multiple sharing` is not a valid Apex keyword.

- D. inherited sharing

`inherited sharing` is also not a valid Apex keyword. While sharing settings can be *inherited* in some contexts (e.g., inner classes), there's no keyword specifically called `inherited sharing`. The default

behavior if no keyword is specified is the same as `with sharing`. So, while closest to correct, it is not the *correct* answer. `with sharing` is the keyword that must be used.

## 32. Question

What is defined here: “IT is a tool that allows defining the relationships between the objects (standard or custom) that take part into the app’s functionality.”

- Data Model
- Objects
- Records
- Schema Builder

**Unattempted**

**Correct:**

A. Data Model

The description “a tool that allows defining the relationships between the objects (standard or custom) that take part into the app’s functionality” accurately defines a *data model*. A data model is a conceptual representation of the data structures and their relationships within a system or application. In Salesforce, this involves defining objects (standard and custom) and the relationships between them (lookup, master-detail, etc.).

**Incorrect:**

B. Objects

Objects are the individual building blocks of the data model (e.g., Account, Contact, Opportunity). While they are *part* of the data model, they don’t define the *relationships* between other objects. The relationships are what constitute the data model itself.

C. Records

Records are instances of objects (e.g., “Acme Corporation” is a record of the Account object). Records contain the actual data, but they don’t define the structure or relationships of the data model. They *populate* the data model.

D. Schema Builder

The Schema Builder is a *tool* within Salesforce that *helps* you visualize and *manage* the data model. It visually displays objects and their relationships. However, the Schema Builder *itself* is not the data model; it’s a tool

for working with it. The data model is the underlying structure, and the Schema Builder is a way to interact with that structure.

### 33. Question

True or False? Data Context is provided to controllers in a Visualforce page by the ID parameter of a page.

TRUE

FALSE

Unattempted

TRUE

The ID parameter in the URL of a Visualforce page *is* how the data context is typically provided to the controller. When you're viewing a record detail page (e.g., /apex/MyVisualforcePage?id=001xxxxxxxxxxxx), the **id** parameter contains the ID of the record. The Visualforce page's controller can then use this ID to query and retrieve the specific record's data, establishing the data context.

### 34. Question

Can Visualforce pages be used to override object actions?

No

Yes

Unattempted

Yes

Visualforce pages *can* be used to override standard object actions (like View, Edit, Delete, etc.). This allows you to replace the standard Salesforce functionality with a custom Visualforce page, giving you complete control over the user interface and behavior.

### 35. Question

SOSL, SOQL or DML: Which one can construct a query that returns specific data?

SOQL

SOSL

DML

Neither

**Unattempted****Correct:** **A. SOQL**

SOQL (Salesforce Object Query Language) is specifically designed for querying Salesforce data and returning specific data based on criteria. It's the primary way to retrieve data from the Salesforce database.

**Incorrect:** **X B. SOSL**

SOSL (Salesforce Object Search Language) is used for text-based searches across multiple objects. While SOSL *can* return data, it's not designed for the same kind of precise, structured querying as SOQL. SOSL is for finding records based on search terms, not retrieving specific fields based on relationships or complex criteria.

 **X C. DML**

DML (Data Manipulation Language) statements (like INSERT, UPDATE, DELETE) are used for *modifying* data, not querying it. DML operations change data in the database, but they don't retrieve data in the way a query does.

 **X D. Neither**

This is obviously incorrect since SOQL is the primary way to query and retrieve data.

### 36. Question

At what level is record type access granted?

- OWDS
- Profile
- Role
- User

**Unattempted****Correct:** **B. Profile**

Record type access is granted at the *profile* level. Profiles control what users can do in Salesforce, including which record types they can access and use. This is a fundamental aspect of profile-based security.

Incorrect:

**X A. OWDS (Organization-Wide Defaults)**

Organization-Wide Defaults (OWDs) control the *baseline* access to records for an object. They set the most restrictive level of access. While OWDs play a role in overall data access, they don't directly control *record type* access. Record type access is *more granular* than OWDs.

**X C. Role**

Roles control access to records based on a hierarchy. While roles *influence* data visibility, they don't directly grant or deny access to specific *record types*. Roles are used in conjunction with OWDs and sharing rules to broaden access, but profiles are what define which record types are available to a user in the first place.

**X D. User**

Record type access is not granted at the individual user level. Instead, it's managed at the *profile* level, and then users are assigned to profiles. This makes administration much more efficient because you manage access for groups of users (via profiles) rather than individual users.

### 37. Question

If a Workflow Rule changes the value of a field into something not validated by a Validation Rule, will the new value be kept?

No

Yes

Unattempted

Yes

If a Workflow Rule (or now, a Flow) changes the value of a field, it *bypasses* validation rules. The new value will be saved, even if it wouldn't normally pass validation. This is a key point to understand about the order of execution in Salesforce.

### 38. Question

If a user wishes for a loop to skip the current iteration and jump to the next, which statement is required?

Continue

Break**Unattempted** A. Continue

The `continue` statement is used within a loop (like `for` or `while`) to skip the *current* iteration and proceed directly to the *next* iteration. The remaining code within the loop's body for the current iteration is not executed.

**39. Question**

Of the following code terms which one responds to the description? “Can assign a left value to a new value expression or an inline SOQL query”

 Variations Operators **A. Assignments** Expressions**Unattempted****Correct:** C. Assignments

The description “Can assign a left value to a new value expression or an inline SOQL query” perfectly defines *assignments*. Assignment statements are used to give a variable (the “left value”) a new value, which can be the result of an expression or the result of a SOQL query assigned directly inline. For instance, `MyVariable__c = 'New Value';` or `List<Account> accounts = [SELECT Id FROM Account];` are both assignment statements.

**Incorrect:** A. Variations

“Variations” isn’t a standard programming term in this context. It doesn’t describe the ability to assign values.

 B. Operators

Operators (like `+`, `-`, `=`, `==`, etc.) are used *within* expressions to perform operations on values. While assignments *use* the assignment operator (`=`), “operators” as a general term doesn’t specifically describe the act of assigning a value.

### ✗ D. Expressions

Expressions are combinations of values, variables, and operators that evaluate to a single value. While an assignment *includes* an expression on the right-hand side, “expressions” as a general term doesn’t specifically describe the act of *assigning* that evaluated value to a variable. The assignment is the action; the expression is what is being assigned.

## 40. Question

How many Master-Detail Relationships are required in a many to many relationships?

- 3
- 1
- 5
- 2

Unattempted

Correct:

✓ D. 2

A many-to-many relationship in Salesforce requires *two* Master-Detail relationships. You create a junction object (also sometimes called a bridge object) that has two Master-Detail relationships: one to each of the objects you want to link in a many-to-many fashion.

Incorrect:

✗ A. 3

Three Master-Detail relationships are not required (or even possible in a standard many-to-many implementation). Two is the specific and correct number.

✗ B. 1

One Master-Detail relationship is insufficient for a many-to-many relationship. A single Master-Detail relationship would create a one-to-many relationship.

✗ C. 5

Five Master-Detail relationships are unnecessary and would not correctly implement a many-to-many relationship. Two is the fixed number required

## 41. Question

As a general rule, declarative tools are suited to automate actions between RELATED or UNRELATED records?

- Neither
- Related
- both related and unrelated records.
- Unrelated

### Unattempted

Declarative tools are generally suited to automate actions between **both** related and unrelated records.

Here's why:

- **Related Records:**
  - Many declarative tools (like workflow rules, process builder, and flows in Salesforce) are designed to easily traverse and manipulate related data. For example, updating child records when a parent record is changed.
- **Unrelated Records:**
  - While relationships simplify automation, declarative tools can also work with unrelated records. For example, a flow could be triggered to update a specific set of records based on criteria, regardless of any direct relationship. Also, with the use of things like scheduled flows, or platform events, unrelated records can be updated.

Therefore, the answer is **Both**.

## 42. Question

Which primitive variable should be used as a flag in Apex to check for a particular condition?

- Blob
- Boolean
- Long
- Integer

### Unattempted

Correct:

- B. Boolean

A Boolean variable is the standard and most appropriate primitive data type to use as a flag in Apex (and most other programming languages). Boolean variables can hold only two values: `true` or `false`, making them ideal for representing conditions or states.

Incorrect:

**X A. Blob**

A Blob is used to store binary data (like files). It's not suitable for representing a simple true/false condition.

**X C. Long**

A Long is a numeric data type used to store large integers. While you *could* use a Long to represent a flag (e.g., 0 for false, 1 for true), it's not the idiomatic or best practice. Boolean variables are far more clear and expressive for this purpose.

**X D. Integer**

An Integer is also a numeric data type, similar to Long but for smaller integers. Like Long, you *could* use it as a flag, but it's not the recommended approach. Boolean variables are specifically designed for true/false values and improve code readability.

### 43. Question

In which section of the DebugLog will the version of the API used be printed?

- Header
- Execution Units
- Code Units
- Log Lines

Unattempted

Correct:

**✓ A. Header**

The API version used for a particular request is included in the *header* section of the debug log. This is one of the first pieces of information logged and is crucial for troubleshooting API-related issues.

Incorrect:

**X B. Execution Units**

The Execution Units section provides performance metrics about different parts of the code execution (methods, triggers, etc.). It does *not* contain the API version information.

### ✗ C. Code Units

“Code Units” is not a standard section name within the debug log.

### ✗ D. Log Lines

The Log Lines section contains the detailed, line-by-line output of the debug statements in your code. While the API version might be *mentioned* somewhere *within* the log lines if you specifically log it, it’s not a standard piece of information displayed in the Log Lines section itself. It’s primarily in the header.

## 44. Question

Are formula fields read only, or can they be edited from the page layout?

- A. Read Write
- B. Encrypted
- C. Read only

Unattempted

Correct:

### ✓ C. Read only

Formula fields are *read-only*. Their values are calculated dynamically based on the formula you define, and users cannot directly edit them on the page layout. The value is automatically updated whenever the underlying data changes.

Incorrect:

### ✗ A. Read Write

This is the opposite of the truth. Formula fields are explicitly *not* read-write. If they were editable, they would no longer be formula fields; they would be regular fields.

### ✗ B. Encrypted

While fields *can* be encrypted, this is a separate concept from whether a field is a formula field. Encryption is about data security, whereas the read-only nature of formula fields is about how their values are determined.

A formula field can technically also be encrypted, but that doesn't change its fundamental characteristic of being read-only.

## 45. Question

What are the two types of Control Flow Statements depending on whether or not conditionality is part of it?

- Conditional Flow Statements and SOQL queries
- SOSL queries and Procedural Loops
- Conditional Flow Statements and Procedural Loops**
- SOQL queries and Procedural Loops

**Unattempted**

**Correct:**

**✓ C. Conditional Flow Statements and Procedural Loops**

Control flow statements in Apex (and most programming languages) can be broadly categorized into two types based on conditionality:

- **Conditional Flow Statements:** These statements (like `if`, `else if`, `else`, `switch`) execute different blocks of code depending on whether a certain condition is true or false. Conditionality is their defining feature.
- **Procedural Loops:** These statements (like `for`, `while`, `do-while`) execute a block of code repeatedly. While loops *can* contain conditional logic *within* them, the loop itself is primarily about repetition, not conditional execution.

**Incorrect:**

**✗ A. Conditional Flow Statements and SOQL queries**

SOQL queries are used to retrieve data from the database. They are not a type of control flow statement. While SOQL can be *used within* control flow statements, they are distinct concepts.

**✗ B. SOSL queries and Procedural Loops**

SOSL queries, like SOQL queries, are for searching data. They are not control flow statements. Procedural loops *are* a type of control flow statement, but SOSL is not.

**✗ D. SOQL queries and Procedural Loops**

Again, SOQL queries are for data retrieval, not control flow. Procedural loops are a type of control flow, but the pairing with SOQL is incorrect in this context.

## 46. Question

In which cases can relationships NOT be created?

- Standard Object to Standard Object
- Custom Object to Standard Object (or vice versa)
- Only custom objects
- Custom Object to Custom Object

### Unattempted

- Between two standard objects **if the relationship type is restricted** by Salesforce (varies by object).
- Or “Standard object to standard object” relationships might be restricted depending on the exact objects involved.

## 47. Question

Are Validation Rules triggered during lead conversion?

- Only if enabled
- No
- Only for standard objects.
- Yes

### Unattempted

D. Yes

Validation rules *are* triggered during lead conversion. This is important because it ensures that the data being created on the resulting Account, Contact, and Opportunity (or any other object you’re converting to) meets the defined data quality standards.

X A. Only if enabled

Validation rules are *always* enforced during lead conversion. There’s no setting to “enable” or “disable” them specifically for lead conversion. They’re part of the standard data validation process.

X B. No

This is the opposite of the truth. Validation rules are a key part of the lead conversion process and are designed to prevent bad data from being created during conversion.

**✗ C. Only for standard objects.**

Validation rules apply to *both* standard and custom objects. So, they'll be triggered during lead conversion regardless of whether you're converting to standard objects (Account, Contact, Opportunity) or custom objects.

## 48. Question

From which menu in setup can users view all COMPLETED apex jobs?

- Apex Triggers
- Apex Jobs**
- Apex Scheduled Jobs
- Apex Classes

**Unattempted**

**Correct:**

**✓ B. Apex Jobs**

The Setup menu option **Apex Jobs** is specifically where you can view the status of all Apex jobs, including completed, failed, and currently running jobs. This is the central place to monitor and manage asynchronous Apex processing.

**Incorrect:**

**✗ A. Apex Triggers**

The Apex Triggers setup menu option is for managing and viewing Apex triggers, which are pieces of code that execute in response to specific events on data. It does *not* show the status of general Apex jobs.

**✗ C. Apex Scheduled Jobs**

The Apex Scheduled Jobs setup menu option is specifically for managing *scheduled* Apex jobs (jobs that run on a defined schedule). While scheduled jobs are a type of Apex job, this menu option doesn't show *all* Apex jobs, only the scheduled ones.

**✗ D. Apex Classes**

The Apex Classes setup menu option is for viewing and managing Apex classes (the code itself). It does *not* display the status of Apex jobs that have been run or are currently running. You manage the *code* in Apex Classes, and you monitor the *execution* of that code in Apex Jobs.

## 49. Question

SOSL, SOQL or DML: Which one of them is used for querying a database?

- A. SOSL
- B. Neither
- C. DML
- D. SOQL

**Unattempted**

**Correct:**

**✓ D. SOQL**

SOQL (Salesforce Object Query Language) is *specifically* designed for querying the Salesforce database. It allows you to retrieve data based on specific criteria and relationships between objects.

**Incorrect:**

**✗ A. SOSL**

SOSL (Salesforce Object Search Language) is used for *text-based searches* across multiple objects. While SOSL *can* return data, it's not designed for the structured, relational querying that SOQL provides. SOSL is for finding records that match certain search terms, not for retrieving specific fields or related data in a structured way.

**✗ B. Neither**

This is incorrect because SOQL is the primary tool for querying the Salesforce database.

**✗ C. DML**

DML (Data Manipulation Language) statements (like **INSERT**, **UPDATE**, **DELETE**) are used for *modifying* data, not querying it. DML operations change data in the database, but they don't retrieve data.

## 50. Question

Which one is NOT one of the four components of the Data Model of Apps in Salesforce?

- Custom Objects
- Schema Builder
- Relationships
- Standard Objects
- Records

### Unattempted

The four components of the Data Model of Apps in Salesforce are:

1. **Standard Objects:** These are predefined objects provided by Salesforce, such as Account, Contact, Lead, Opportunity, etc.
2. **Custom Objects:** These are objects created by users to store specific data related to their business needs.
3. **Relationships:** These define how different objects are connected to each other, such as a one-to-many relationship between an Account and its Contacts.
4. **Records:** These are individual instances of an object, representing specific data points.

Therefore, **Schema Builder** is not one of the four components of the Data Model of Apps in Salesforce.

Schema Builder is a tool used to visually design and manage the relationships between objects in your Salesforce org.

## 51. Question

Which method annotation can be used to make a private class visible to a unit test?

- Both
- @isTest
- Neither
- @testVisible

### Unattempted

Correct:

- D. @testVisible

The `@testVisible` annotation is specifically designed to allow access to private members (variables, methods, or inner classes) of a class from a unit test class. This is essential for testing code that would normally be inaccessible due to its private scope.

Incorrect:

**X A. Both**

While `@isTest` is *related* to testing (it marks a class as a test class), it doesn't, by itself, grant access to private members. You need `@testVisible` *in addition* to `@isTest` for this purpose.

**X B. `@isTest`**

As explained above, `@isTest` marks a class as a test class, but it does *not* provide access to private members. You still need `@testVisible` on the specific private element you want to access.

**X C. Neither**

This is incorrect. `@testVisible` is precisely the mechanism provided by Salesforce for this purpose.

## 52. Question

Which of the primitive data types responds to this definition? "Store a true or false value"

- Blob
- Integer
- Boolean
- Double
- Date
- Decimal

**Unattempted**

**Correct:**

**✓ C. Boolean**

Boolean is the primitive data type specifically designed to store true or false values. It's the most straightforward and appropriate choice for representing conditions or flags.

**Incorrect:**

**X A. Blob**

Blob is used for storing binary data, like the contents of a file. It's not related to true/false values.

**X B. Integer**

Integer stores whole numbers. While you could technically use integers to represent true/false (e.g., 0 and 1), it's not the standard practice, and Boolean is much clearer.

#### ✗ D. Double

Double stores floating-point numbers (numbers with decimal points). It's used for numerical values, not logical true/false values.

#### ✗ E. Date

Date stores a date value (year, month, day). It's for representing calendar dates, not true/false conditions.

#### ✗ F. Decimal

Decimal stores numbers with a specific precision and scale (useful for financial calculations). While related to numbers, it's not used for true/false values.

### 53. Question

What is the CSS file of a lightning component responsible for?

- There is no CSS file on the lightning component bundle.
- To style the component and match a specific style.
- It defines how the component UI will react to client events.

Unattempted

Correct:

- B. To style the component and match a specific style.

The CSS file (or more accurately, the styling resources) within a Lightning Web Component (LWC) or Aura component bundle is responsible for defining the *visual presentation* and styling of the component. It controls things like colors, fonts, layout, and overall appearance.

Incorrect:

- A. There is no CSS file on the lightning component bundle.

This is incorrect. While a CSS file isn't *strictly required*, it's the standard and recommended way to style Lightning components. You can use CSS within your components to control their appearance. It's a best practice.

- C. It defines how the component UI will react to client events.

How a component reacts to client events (like clicks, mouseovers, etc.) is handled by JavaScript within the component's controller (for Aura components) or within the component's JavaScript file (for LWC). CSS is only for styling; it does not handle event logic. JavaScript handles the behavior.

## 54. Question

Are Roll-Up Summaries and quick actions considered declarative automation tools?

- Not Roll-Up summaries.
- Yes
- No, neither are.
- Not Quick Actions.

Unattempted

Correct:

- B. Yes

Both **Roll-Up Summary fields** and **Quick Actions** are considered **declarative automation tools** in Salesforce because they allow automation without writing code:

- **Roll-Up Summary Fields:** Used to aggregate values (SUM, COUNT, MIN, MAX) from child records in a **master-detail** relationship, automating data calculation without Apex code.
- **Quick Actions:** Allow users to create, update, or log a call on a record with minimal input, improving efficiency through declarative automation.

Incorrect:

- A. Not Roll-Up summaries. — Roll-Up Summary fields **are** declarative automation tools as they automate data aggregation in a **point-and-click** manner.
- C. No, neither are. — Both **Roll-Up Summaries** and **Quick Actions** are declarative automation tools, making this statement incorrect.
- D. Not Quick Actions. — Quick Actions **are** declarative automation tools because they allow users to perform actions efficiently without code.

## 55. Question

Which one can be run by Visual Studio Code?

- Both
- Apex Test Suites

Neither Apex Tests**Unattempted**

Based on the information available, Visual Studio Code, with the Salesforce Extensions, provides robust capabilities for running Apex tests. This includes:

- **Apex Tests:**
  - Visual Studio Code allows developers to run individual Apex test methods, test classes, and all Apex tests within a project.
  - The VS Code interface provides features for viewing test results, including pass/fail status and code coverage.
- **Apex Test Suites:**
  - Visual Studio Code, in conjunction with the Salesforce CLI, supports the execution of Apex Test Suites. This enables developers to group and run collections of Apex tests.

Therefore, the answer is: **Both**

Here's a breakdown:

- VS Code with the Salesforce extensions is designed to facilitate Apex development, and that includes comprehensive testing functionality.
- The test explorer within VS code, and the use of the Salesforce CLI from within VS code allows for the running of single tests, and test suites.

**56. Question**

Would you use Declarative or Programmatic development if there is a high level of complexity?

 Programmatic Declarative**Unattempted**

Correct:

 A. Programmatic

For high levels of complexity, programmatic development (Apex, Visualforce, Lightning components) is generally the better choice. While declarative tools are powerful, they have limitations when dealing with intricate logic, complex calculations, or highly customized integrations. Programmatic development provides the flexibility and control needed to handle complex scenarios.

Incorrect:

### ✗ B. Declarative

While declarative tools are excellent for many automation tasks, they are not well-suited for *high* complexity. Trying to implement very complex logic using only declarative tools can become cumbersome, difficult to maintain, and may even hit platform limitations. For complex scenarios, you'll almost always need the power and flexibility of programmatic development.

## 57. Question

Which of the three frameworks is the ONLY one that can make use of Application Events to publish and subscribe to events?

- Apex Classes
- Lightning Components
- Aura Components

**Unattempted**

**Correct:**

- C. Aura Components

Aura Components are the *only* of the three options listed that can directly use Application Events for publish-subscribe communication. Application Events are a feature specific to the Aura framework.

**Incorrect:**

- A. Apex Classes

Apex classes cannot directly publish or subscribe to Aura Application Events. Apex can publish Platform Events (which are different), but not Aura Application Events.

- B. Lightning Components

Lightning Web Components (LWC) *cannot* use Aura Application Events. LWC uses a different event system (DOM events and the Lightning Message Service) for component communication. LWC is the newer framework, and while it shares some concepts with Aura, Application Events are not one of them.

## 58. Question

Does Apex Classes or Interfaces require the interface keyword?

- Neither
- Both

**Interfaces** **Classes****Unattempted****Correct:** **C. Interfaces**

Interfaces in Apex *require* the `interface` keyword. This is how you define an interface. For example:

```
public interface MyInterface { ... }
```

**Incorrect:** **A. Neither**

This is incorrect. Interfaces *must* use the `interface` keyword.

 **B. Both**

Classes do *not* use the `interface` keyword. Classes use the `class` keyword. For example:

```
public class MyClass { ... }
```

 **D. Classes**

Classes use the `class` keyword, not the `interface` keyword. The `interface` keyword is exclusively for defining interfaces.

**Use Page numbers below to navigate to other practice tests**

Pages: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#)

We help you to succeed in your certification exams

We have helped over thousands of working professionals to achieve their certification goals with our practice tests.

Skillcertpro

Quick Links

[ABOUT US](#)

FAQ

BROWSE ALL PRACTICE TESTS

CONTACT FORM

## Important Links

[REFUND POLICY](#)

[REFUND REQUEST](#)

[TERMS & CONDITIONS](#)

[PRIVACY POLICY](#)

[Privacy Policy](#)