



## Appian ACD101 Practice Questions

Shared by Conrad on 08-10-2026

**For More Free Questions and Preparation Resources**

Check the Links on Last Page



## Question 1

---

Question Type: MultipleChoice

---

You are reviewing a timesheet submission on a form.

The buttons on the form are: "Cancel", "Deny", and "Approve". Most of the time, users will approve the timesheet.

According to Appian best practices, which button styling is most appropriate on the button labelled "Approve"?

Options:

- A- Primary
- B- Secondary
- C- Destructive

Answer:

A

## Question 2

---

Question Type: MultipleChoice

---

How can you add test data into your rule inputs while editing an interface object?

Options:

- A- Select the Performance tab and set test values.
- B- Select the Test button and set test values.
- C- Select the Gear icon, select Properties, and set test values.

Answer:

B

Explanation:

While editing an interface object in Appian, you can add test data into your rule inputs by

selecting the Test button. This feature allows you to simulate how the interface would behave with specific inputs, facilitating a more efficient and accurate design and debugging process. Reference:

Appian Documentation: Testing Interfaces

### Question 3

---

Question Type: MultipleChoice

---

A customer wants to display a small toolbar with three icons - a "sad face," a "face with neutral expression," and a "happy face" - on the bottom of every page in their application.

Users will be instructed to use the icon that best expresses their current experience using the application. This will allow the customer to collect valuable data about users.

Which object type should be called from each page to implement this feature?

#### Options:

---

- A- An interface, because the component must render user interface elements.
- B- An expression rule, because the component captures expressions of user sentiment for analysis.
- C- A decision, because the component captures a choice that users select from an array of custom selection components, rather than a standard dropdown or radio button.

#### Answer:

---

A

#### Explanation:

---

In Appian, interfaces are used to design and render user interface elements, allowing for the creation of custom layouts and components that can interact with users. In this scenario, the toolbar with emotion icons is a UI element that needs to be consistently presented across various pages. Using an interface ensures that these icons can be interactively used by users to express their sentiments, and the data collected can be analyzed for user experience insights. Interfaces are capable of embedding such UI elements and can be invoked or included across multiple pages within an application, making them the ideal choice for this requirement.

Reference:

Appian Documentation on Interfaces: Provides detailed guidance on how to design and use interfaces to create user-centric UI components.

## Question 4

---

Question Type: MultipleChoice

---

Which option best describes the main purpose of the interface design object?

Options:

- A- Provides a method for an Appian application to interact with an external application or service.
- B- Provides a method for end users to interact with an Appian application.
- C- Provides a method for an Appian application to interact with a database.

Answer:

B

Explanation:

The primary purpose of the interface design object in Appian is to provide a method for end users to interact with an Appian application. Interfaces are the user-facing components that allow for the input, display, and manipulation of data within the application, facilitating user engagement and task completion. Reference:

Appian Documentation: Interface Design

## Question 5

---

Question Type: MultipleChoice

---

Which step can be critical in passing information from a form back to a process model?

Options:

- A- Configure the Data Management tab.
- B- Configure the activity class parameters of a Write to Data Store Entity node, a
- C- Configure inputs on the Data tab of a User Input Task.

Answer:

---

C

Explanation:

---

The critical step in passing information from a form back to a process model is to configure inputs on the Data tab of a User Input Task. When you create a User Input Task, it includes a form for users to interact with. The data entered into this form can be mapped to process variables via the Data tab configuration. This ensures that the information collected in the form is available to the process for further use. Reference: Appian Documentation - User Input Tasks



## Question 6

---

Question Type: MultipleChoice

---

You created and published a new process model.

The process model has a start form with two synchronous subprocesses with 40 and 66 nodes each. All nodes are chained from the start node through the subprocesses to the end node. After the tasks and subprocesses, there is a second User Input Task in which the user can confirm the entries and add a comment.

When testing as a normal Acme business user, you see that the confirmation screen is not shown to you.

What might be the reason for this behavior?

Options:

---

- A- The maximum number of activity chained nodes is exceeded and breaks.
- B- The second User Input Task is assigned to the process initiator.
- C- The second User Input Task is assigned to the Acme business user group.

Answer:

---

A

Explanation:

---

In Appian, there is a limitation on the number of activities that can be chained in a process, known as the 'chaining limit.' If a process model exceeds this limit, which includes synchronous

subprocesses and their nodes, the process may break or not behave as expected. In this scenario, with two large subprocesses chained from start to end, the maximum number of activity chained nodes could be exceeded, resulting in the confirmation screen not being shown. Adjusting the process model to reduce chaining or using asynchronous patterns where possible can help mitigate this issue. Reference: Appian Documentation - Process Model Best Practices

## Question 7

---

Question Type: MultipleChoice

---

Which statement about local variables is valid?

### Options:

---

- A- The data type of a local variable is determined by its value.
- B- Local variables can only store primitive data types, such as numbers and strings.
- C- Local variables must have an initial value set when defining them.

### Answer:

---

A

### Explanation:

---

In Appian, the data type of a local variable is inferred from the value it is set to. Unlike some other programming languages where the data type must be explicitly declared, Appian determines the data type automatically based on the initial value assigned to the local variable. Local variables in Appian are quite flexible and can store various types of data, including complex data types, not just primitive ones. Reference: Appian Documentation - Local Variables

## Question 8

---

Question Type: MultipleChoice

---

You have a record action that should only be visible to certain users under conditions specified by an expression.

How should you configure this?

**Options:**

---

- A- Set security on the process model.
- B- Set permissions directly on the user object.
- C- Set security on the record action.

**Answer:**

---

C

**Explanation:**

---

To control the visibility of a record action based on certain user conditions or expressions, the security settings of the record action itself must be configured. This involves defining conditions or expressions within the record action's security settings that evaluate whether a user meets the criteria to see and interact with the action. This method allows for dynamic control over the accessibility of actions based on user roles, specific attributes, or other contextual factors, ensuring that only authorized users can see and execute the action under specified conditions.

Reference:

Appian Documentation on Record Actions: Details how to configure and secure record actions, including visibility and permissions, to tailor the user experience within record views.

---

**Question 9**

---

**Question Type:** MultipleChoice

---

What are three ways to trigger a process from a process model? (Choose three.)

**Options:**

---

- A- Use a subprocess.
- B- Call a web API.
- C- Use a!recordActionItem().
- D- Use a Start Process smart service.
- E- Run a!startProcess() from a script task.

**Answer:**

---

A, D, E

### Explanation:

---

There are multiple ways to trigger a process from a process model in Appian:

A subprocess is a process model that is called from within another process model.

A Start Process smart service is a flow element that you can use within a process model to start another process.

The `startProcess()` function can be used within a script task to start a process programmatically. These methods allow a developer to initiate a new process instance from an existing process. Reference: Appian Documentation - Process Model Triggers and Smart Services



### Question 10

---

Question Type: MultipleChoice

---

Review the following code snippet:

```
index({'a', 'b', 'c'}, 1, 'x')
```

Which value is returned?

### Options:

---

- A- a
- B- b
- C- c
- D- x



### Answer:

---

D

### Explanation:

---

The `index()` function in Appian is used to replace an element at a specified index in a list. In the code snippet `index({'a', 'b', 'c'}, 1, 'x')`, the function is called to replace the element at index 1 of the list `{'a', 'b', 'c'}` with the string `'x'`. Since Appian lists are 1-indexed, this means that the first element `'a'` will be replaced by `'x'`. However, since the function is used to modify a list and not to return a value, if this function is used as is without being part of a larger expression that outputs the list, it would not return any of the listed values. Reference: Appian Expression

## Language Documentation - Functions



To Get Premium Files for ACD101 Visit

<https://www.p2pexams.com/products/acd101>

For More Free Questions Visit

<https://www.p2pexams.com/appian/pdf/acd101>

**20%**  
**DISCOUNT**

**P2P**  
exams