PLEASE NOTE: Not all answers are provided in this document (e.g. Answers to question asking for your personal opinion, or questions asking you to predict what will happen when an action is performed).

Module 1.........................................................................................................................2
  Lab 1.1 Using the Syntax Editor .....................................................................................2
  Lab 1.2 Syntax Checking .................................................................................................2
  Lab 1.3 Explore Scripting Resources ...............................................................................3

Module 2.............................................................................................................................4
  Lab 2.1 Two Simple Client Scripts .................................................................................4
  Lab 2.2 g_form and g_user .............................................................................................4
  Lab 2.3 Client Script Debugging ......................................................................................5
  Lab 2.4 Client Scripting for Mobile ................................................................................6
  Lab 2.5 Client Scripting with Reference Objects ............................................................6

Module 3.............................................................................................................................7
  Lab 3.1 Incident Resolved/Closed UI Policy ...................................................................7

Module 5.............................................................................................................................7
  Lab 5.1 Business Rule Debugging ..................................................................................7
  Lab 5.2 Current and Previous .........................................................................................8

Module 8.............................................................................................................................8
  Lab 8.1 Tracking Impersonations ....................................................................................8

Module 9.............................................................................................................................9
  Lab 9.1 Classless Script Include .....................................................................................9
  Lab 9.3 HelloWorld GlideAjax .......................................................................................9

Module 10...........................................................................................................................9
  Lab 10.1 Incident SLA Workflow .....................................................................................9
Module 1

Lab 1.1 Using the Syntax Editor

B9 - QUESTION: Turn off syntax highlighting by selecting the Toggle Syntax Editor button. What differences do you notice in the Syntax Editor?

ANSWER: The Syntax Editor turns into a basic text field. Standard code editing features are no longer available.

C7a - QUESTION: Review the description for the 'try' macro. Where did the value come from?

ANSWER: The Comments field in the Editor Macro record.

C7b - QUESTION: Review the description for the 'info' macro. Why is it blank?

ANSWER: The Comments field in its Editor Macro record is empty.

Lab 1.2 Syntax Checking

A5 - QUESTION: Try to save your script. Did it save? How can you tell?

ANSWER: The Script field’s label flashes yellow and an error message identifying issues appears below the field.

A9 - QUESTION: Did the Syntax Checker find all the errors in the script? Which error(s) were not found?

ANSWER:
- Line 3: Missing quotes around each item in the array. Not “required” but considered best practice.
- Line 3: Extra comma at the end of the array.
- Line 7: Invalid reference to the ‘peripheral’ array, it should read ‘peripherals’ to match the variable name declared on line 3.
Lab 1.3 Explore Scripting Resources

A4 - QUESTION: Where do you search for existing content in Community?
ANSWER: Use the Search (magnifying glass) icon at the top-right corner of the page.

A6 - QUESTION: When you cannot find the information you are looking for in Community; how do you ask a question?
ANSWER: There are a number of places on Community you can ask a question.

- The Search Content, Forums and People input field on the main page.

- After you have logged in, the Ask input field in a specific community (e.g. Communities > Developer > Scripting and Coding).

- Reply to an existing post to clarify any additional questions you may have on the topic.

B5 - QUESTION: Review the API release notes article. What information is provided?
ANSWER:
- New scoped classes and additional methods to existing classes.
- New global classes and additional methods for existing classes.

C8 - QUESTION: What Date/Time format does the GlideDateTime object use?
ANSWER: Greenwich Mean Time (GMT).
Module 2

Lab 2.1 Two Simple Client Scripts

B1 - QUESTION: Open any Incident record. What happens when the form loads?

ANSWER: The alert message box displays advising “The form has finished loading and is ready for user input”.

B6 - QUESTION: Does the Lab 2.1 onLoad Alert Client Script execute again? Why or why not?

ANSWER: Yes. The Client Script’s trigger is set to execute onLoad, therefore the alert will display every time the form loads.

B7 - QUESTION: Create a new Incident. What happens when the form loads for the new incident?

ANSWER: The Client Script’s trigger is set to execute onLoad, therefore when a new Incident opens, the form loads.

C6 - QUESTION: Did the “Resolved” or “Closed” values remain in the State field? Why or why not?

ANSWER: The script set the value of the saveAndClose variable to false. This prevented the record from saving any changes when the callback function executed. The script will only save the record if the saveAndClose variable contains the value true.

Lab 2.2 g_form and g_user

C4 - QUESTION: What happened to the Incident form when you cancelled the submission? Was the Incident submitted?

ANSWER: Two information messages appear at the top of the form and four field messages appear below the fields identified in the showFieldMsg() methods used in the script (category, cmdb_ci, assignment_group, short_description). The JavaScript confirm() method returns false when “Cancel” is selected, this stops the submission of the Incident record.

C5 - QUESTION: Modify the confirmation dialog box in the script so confirmation message includes the logged in user's first name. Which property do you need to use?

ANSWER: g_user.firstName

C9 - QUESTION: Create a new Priority-1 Incident and Save (not Submit) the record. Did the confirmation message display? Explain why not?

ANSWER: The script first checks to determine if the logged in user creating the record does NOT have the major_inc_mgr. If yes, the script does not execute.
Lab 2.3 Client Script Debugging

E5 - QUESTION and ANSWER: Review the log messages in the JavaScript Log.
- What is the value currently in the ‘incState’ variable? The variable is empty
- What should it be? 1
- Does the string “LINE 11 EXECUTED” appear as a logged message? No

E6 - QUESTION: What can you conclude regarding the debugging output currently in the JavaScript Log?
ANSWER: Something is not right with the incState variable and because the JavaScript Log does not contain the message regarding Line 11 executing, the error occurs prior to this line of code in the script.

E7 - QUESTION: At this point, you know something is not right with the value incState variable. Before leaving the Incident form, double-check the State field’s name by right-clicking the field’s label. What is the exact spelling of the State field's name?
ANSWER: state

E9 - QUESTION: Review the State field’s name in the “g_form.getValue('incident');” statement, does it match the field name you documented in step-7?
ANSWER: No, the State field’s name is not correct, it should read ‘state’ not ‘incident’.

E19 - QUESTION: Review the InfoMessage at the top of the form. Are you the only person who can see this script output?
ANSWER: No, everyone can see it.

E20 - QUESTION: Form messages are a good debugging strategy as the results are instantly presented at the top of the form you are testing on. Would this type debugging strategy be best in a development or production instance?
ANSWER: Development, the statement should be removed before moving the script to a production instance.

G2 - QUESTION: Can Beth turn the JavaScript Log and Field Watcher on? Why or why not?
ANSWER: No she cannot, she does not have the ‘admin’ role, which is required to use the feature.

G4 - QUESTION: Can Fred turn the JavaScript Log and Field Watcher on? Why or why not?
ANSWER: Yes, he can. Fred has the ‘admin’ role, which is required to use the feature.
G5 - QUESTION: Based on this test, what can you conclude regarding the JavaScript Log and Field Watcher while you are impersonating another user?

ANSWER: Regardless of who you are impersonating, that user must have the ‘admin’ role to use the JavaScript Log and Field Watcher.

**Lab 2.4 Client Scripting for Mobile**

D3d - QUESTION: Did the Assigned to label flash? Why or Why not?

ANSWER: The label did not flash. The `flash()` method is not available on the mobile platform.

**Lab 2.5 Client Scripting with Reference Objects**

B5 - QUESTION: Are the *Priority*, *Risk*, and *Impact* fields read-only? Should they be?

ANSWER: They should not be read-only as the script does not currently include commands to set the read-only property to `false` in the event the CI name is NOT 3D Pinball.
Module 3

Lab 3.1 Incident Resolved/Closed UI Policy

B11 - QUESTION: Does the ‘Reverse if false’ field need to be selected in order for the ‘Execute if false’ script to execute?

ANSWER: Yes, it does. The script in the Execute if false field will never execute if the Reverse if false field is not selected.

B12 - QUESTION: When the ‘Reverse if false’ field is selected and the ‘Execute if false’ field is empty, does the reverse of what is scripted in the ‘Execute if true’ field occur?

ANSWER: No, it does not. If the Reverse if false field is selected and the Execute if false field is empty, a script will not execute if the UI Policy’s conditions are not met.

D3 - QUESTION: The Info Message does not appear on the mobile device. How come?

ANSWER: The default value of a UI Policy’s UI type is Desktop. It must be updated to All for it to execute on all device types.

Module 5

Lab 5.1 Business Rule Debugging

B5 - QUESTION: What can you conclude about where breakpoints can be set?

ANSWER: Breakpoints can be set in any scriptable field with the full editor as well the script field in the debugger itself.

B8 - QUESTION: Script execution is paused at the first breakpoint. How can you tell?

ANSWER: The number in the gutter and the line of code are highlighted red.

B12 - QUESTION: Explain why the ‘myNum’ variable value is sometimes undefined and other times contains a value?

ANSWER: When the Script Debugger pauses at a breakpoint, it stops the script before the statement it is paused at executes. In this lab, the first breakpoint paused the script before the value of the myNum variable was set, this is why it is undefined. Once the script continued to the next breakpoint, the value of the variable was set to the value current in the State field.

C6 - QUESTION: Document the “previous” object's short_description field value, document the “current” object's short_description field value. Why are they different?

ANSWER: current is an object that stores the current record’s fields and values, previous is an object that stores the record’s fields and values before any changes are made.
D11 & D14 - QUESTION: How can you be sure which script you are currently debugging?
ANSWER: The Code Pane Header reads Business Rule > Lab 5.1 Business Rule Debugging.

E5 - QUESTION: Which undefined function produced an error and why?
ANSWER: The thisFunctionDoesNotExist() function produced an error because it was scripted inside a try/catch. The error handling scripted in the catch block uses the global gs.log() method to produce the error output.

E6 - QUESTION: Was a log message produced for the undefined thisFunctionAlsoDoesNotExist() function?
ANSWER: NO

F3 - Business Rule Debugging” Business Rule by searching for the string == Lab 5.1 Business Rule Debugging. Did your test meet the Business Rule's Condition criteria? How can you tell?
ANSWER: It did not meet the criteria and the debugging output advises it skipped the Business Rule because the condition current.state != 17 was not satisfied. An excellent strategy to see if your Business Rule actually executed.

Lab 5.2 Current and Previous

C6 - QUESTION: Does the RCA included field need to remain visible on the form?
ANSWER: It does not. The Business Rule will take care of ensuring it is populated properly. Manual updates are not needed. The field can then be used to query the table for a list of records with RCA details included.

Module 8

Lab 8.1 Tracking Impersonations

C8 - QUESTION: Why are there only two ‘impersonation.start’ events?
ANSWER: You are logged in as the System Administrator (user id: admin), that is the user performing the impersonations. The System Administrator impersonated two people (Beth and Fred), this is the reason there are only two ‘impersonation.start’ events.
Module 9

Lab 9.1 Classless Script Include

C7 - QUESTION: After setting this.debug = false, does a new Incident's properties and values appear in the list of Script Log Statements?

ANSWER: No. The ‘if’ statement in the Script Include first checks to determine if the value of this.debug is equal to “true”. If yes, the script will write a log message containing the debugPrefix property and the passed in string. If no, (which it is not at this point in the lab as this.debug was just updated to equal false), no further script logic executes.

C8 - QUESTION: Explain why a Script Include to write log data with an "on/off" parameter is useful for debugging scripts?

ANSWER: It is another quick and easy way to review the values of every field in the record. The on/off parameter makes activate and de-activate the creation of the logs very easy and if you save the Script Log Statements module as a favorite, getting to the logs files is quite quick too.

C10 - QUESTION: Do you need to make the Logging Script Include inactive? Why or why not? Is there a performance impact if it remains active?

ANSWER: No, you do not need to make the Script Include inactive. It only runs when called, otherwise it does nothing. As long as the Business Rule is marked inactive, the Script Include will not execute.

Lab 9.3 HelloWorld GlideAjax

B5 - QUESTION: The first parameter in the script is 'sysparm_name', it is a reserved parameter name. What information does it pass to the Script Include?

ANSWER: sysparm_name pass the name of the method/function you want to use.

B6 - QUESTION: Other than beginning with a “sysparm_” prefix, do additional parameters have to use reserved parameter names?

ANSWER: They do not. As long as the parameter name begins with sysparm_ and has no spaces or special characters, you can give the parameter whatever name you like. It is considered best practice for parameter names to describe the data being passed to the Script Include (e.g. sysparm_start_date, sysparm_user_name, sysparm_manufacturer).

Module 10

Lab 10.1 Incident SLA Workflow

D1 - QUESTION: Do you have to publish the Workflow to test its functionality? Why or why not?

ANSWER: No, you do not have to publish the Workflow to test it. The checked-out version of the Workflow will execute for the person who has it checked out.