

Review for Unit 5 Assessment 1

1-

Content: Unit 5 Lesson 1

An "Event-driven program" is a program that runs in relation to an event, such as user actions (mouse clicks, button presses, etc.) Depending on what the event is, multiple functions can be run in a certain order (as an algorithm) to execute a task. Or, a program can run with a specific flow depending on the event. This flow can be determined with a written algorithm.

For example, a program is written to fix the size of an image by making it bigger with a click of a button. When the button is clicked, there can be code inside the event block that implements an algorithm that adds 10 pixels onto the current image.

2-

Content: Unit 5 Lesson 1

An element must be given a unique ID so a program can reference it; however, it does not need an event handler (lines of code that is executed due to an event such as mouse clicks, keyboard presses, and so forth).

For example, a screen has a button and an image on it. Both the button and the image have unique element IDs (such as "button1" and "image1"). The button has an event handler because the user would need to click it for code to run; however, the image does not have an event handler because it displays an image without needing an action from the user and thus, does not necessarily have an event handler associated with it.

3-

Content: Unit 5 Lesson 4

Debugging a program by printing out the current values with `console.log` and seeing what the program is internally doing at a certain moment will be helpful. Asking for help from another person is also a good way of debugging, especially if you've been working on your code for a long time and need a fresh pair of eyes.

On the other hand, changing variable names will not help because you are only renaming the problem rather than finding and fixing the problem. Computers read code line by line, so rewriting chunks of code is the same as rewriting the same problem.

4-

Content: Unit 5 Lesson 2

What to look for:

- A button and an image are both UI elements.
- Both have positions.
- Both can have background images.
- Both can respond to the same set of events. For example, if a programmer has written an event handler for when a button is clicked, she could write an analogous event handler to perform the same action when an image is clicked (or even just switch the ID of the event handler from the id of the button to the id of the image).

5-

Content: Unit 5 Lesson 4

Solve this problem by going through the lines of code while keeping track of the value of each variable. It is important to note that if a variable is assigned a new value, the lines of code that reference that variable afterwards also references the new value.

- At the very beginning, $a = 0$, $b = 3$, $c = 4$
- $a = a + c$ which equals $0 + 4 = 4$. Thus, variable "a" is changed to the value 4
- *Now, the value of "a" is set to 4*
- $b = a + c$ which equals $4 + 4 = 8$. Variable "b" now equals 8
- $c = a + c$ which equals $4 + 4 = 8$
- By the end of the program, $a = 4$, $b = 8$, and $c = 8$

6-

Content: Unit 5 Lessons 4,5

Since Jasmine wants to increase the `cartTotal` by one every time the "addItemButton" is clicked, the new `cartTotal` should be the current `cartTotal` plus one. In other words,

`cartTotal = cartTotal + 1;`

Since the variable `cartTotal` is already initialized in line 1, the word "var" is not necessary because that would create a new local variable instead of adding on to the existing `cartTotal`.

7-

Content: Unit 5 Lesson 4

General idea: a temporary variable must be made to contain one of the existing information before switching the variable information. Switching "to = from" and then "from = to" is incorrect because both variables would then equal "from".

- A "temp" variable is created and points to one value, "from"
- Since "from" value is stored within "temp", it is safe to write "from = to"
- "to" needs to point to the value of "from" BUT cannot be "to = from" because at this point, "from" stores the value of "to"
- Therefore, "to = temp" because "temp" holds the original value of "from"

Note:

- It may be helpful to draw out boxes containing "Bruce Wayne" and "James Bond", and change the arrows "from", "to", and "temp" to visualize the problem.
- Optional: For further understanding, read up on the term "reference semantics"

8-

Content: Unit 5 Lesson 4

Strategy: Go through the algorithm and keep track of the changes for each line

- Start "Alexander Graham Bell was born 141 years before Benjamin Franklin, so he was never able to telephone his neighbors."
- Change all "Benjamin Franklin" to "apple"
- Current: "Alexander Graham Bell was born 141 years before apple, so he was never able to telephone his neighbors."
- Change all "apple" to "Alexander Graham Bell"
- Current: "Alexander Graham Bell was born 141 years before Alexander Graham Bell, so he was never able to telephone his neighbors."
- Change all "Alexander Graham Bell" to Benjamin Franklin"
- Final: "Benjamin Franklin was born 141 years before Benjamin Franklin, so he was never able to telephone his neighbors."

The algorithm is incorrect because the last two steps should be switched.

9-

Content: Unit 5 Lesson 4 -- requires transfer/application of turtle and variable assignment from mathematical expression (in lesson 4)

Going through the code line by line:

- 1: variable "dist" is set to 100
- 2: variable "radius" is set to $\text{dist}/20 = 5$
- 3: The pen is put down to start drawing
- 4: a for loop is initiated and will repeat the code inside it 4 times
- 5: move forward 100 steps
- 6: draws a dot with radius of 5
- 7: turns 90 degrees to the right

After lines 5-7 are repeated four times, a square is drawn with a small circle at each corner. Note that the dots drawn are a lot smaller than the length of the square.

10-

Content: Unit 5 Lesson 4

Solve this problem by going through the lines of code while keeping track of the value of each variable. It is important to note that if a variable is assigned a new value, the lines of code that reference that variable afterwards also references the new value.

- At the very beginning, $a = 3$, $b = 6$, $c = 10$
- $a = b / a$ which equals $6 / 3 = 2$. Thus, variable "a" is changed to the value 2
- *Now, the value of "a" is set to 2*
- $b = c - a$ which equals $10 - 2 = 8$. Variable "b" now equals 8
- $c = b / a$ which equals $8 / 2 = 4$
- By the end of the program, value of $c = 4$

11-

Content: Unit 5 Lesson 4 -- requires application of variable assignment via math expression + transfer of turtle drawing.

Since the student wants each dot to be half the radius of the previous dot, this means the next dot's radius is multiplied by one half of the dot below it. Therefore, "middle" will be one half of the bottom dot and the "top" will be one half of the middle dot. In other words,

- `var middle = bottom/2;`

- `var top = middle/2;`

The main concept of this problem is that the student only wants to change the radius of the bottom dot, and since the next dot is half of the previous, the variables "middle" and "top" can be changed accordingly.

12-

Content: Unit 5 Lesson 5 - application of local scope and debugging

- The function "fiveMore()" is called.
- Inside this function, a new variable "x" is set to 5
- The computer exits this function and moves on to line 5 where a variable y is created
- The value of y is set to "3 + x" **However**, the variable x was initiated locally inside "fiveMore()" and is not accessible outside of the function. Therefore, there is an error because the variable x does not exist outside the method.

13-

Content: Unit 5 Lesson 5 - application of local v. global

Going through each line of code:

- 1: A variable x is set to the value 10
- 2: The function "increase()" is called where a separate local variable that is also named "x" is set to the value 5
- 3: Exiting the function "increase()", in line 3 the value of x is still 10 because the local variable inside "increase()" is not accessible outside outside the function. Thus, in line 3, $x = x + 3$ which equals $10 + 3 = 13$
- Printing the value of x, the computer outputs "13"

14-

Content: Unit 5 Lesson 5

A global variable, often declared at the top of a program, can be accessed and changed by any function within the program. A local variable, however, only exists within the function in which it was declared. A global variable should be used any time the programmer needs access to it from more than one function. For example, if I were to make a clicker game and wanted the

number of successful clicks in a row to be shown on the user's screen, I would have to create a global variable for score.

What to Look For:

- Student describes the difference between a local and global variable.
- Student explains why a global variable may be necessary
- Student uses an example to support their explanation.