**Mini-Lab: A Whirl of Color**

Stop and Think Questions

*About Face!*

Which statements in Example 1 will be executed when there are no lemon squares? In what order will they be executed? Which statements will be executed, and in what order, when there *are* lemon squares?

In Example 1, if there are no lemon squares, the only actions that will be executed are takeASandwich(), takeCoffee(), and sitDownAndEat() and they will be executed in that order. When there are lemon squares in Example 1, the student will first takeASandwich(), then takeCoffee(), then takeALemonSquare(), and finally sitDownAndEat().

Where is the statement that sets the dimensions of the aquarium? In what class and what method?

The statement that sets the dimensions are in the main method of the AquaSimApplication class and that statement creates an object reference to the Aquarium class.

You could check whether to reverse direction or not and then move forward, or you could move forward first and then check whether to reverse direction. Does the order matter? Consider three cases: a) for a fish that was constructed along the left wall, b) for a fish that was constructed in the middle of the aquarium, and c) for a fish that was constructed along the right wall. Now ask yourself again: Does the order matter?

The order does matter because in case c) if the fish were to be constructed along the left wall then the if statement would be true and therefore change the fishes direction, but then the fish will be facing the wall and won’t be able to move forward till it changes direction again. The simulation would still run, but it would be better to move forward first and then check to change direction.

*One Fish, Two Fish, Red Fish, Blue Fish*

Predictions for results:

The fish will be either blue or red and will have a new random selection each time the program runs. There is no way to know which fish will be which color.

The results were as predicted.

*Rainbow Fish*

Predictions for results:

I predict that the fish will be a random mix of the 6 different colors but not every color will appear each time because there are only 3 fish. There is no way of knowing the fish color still.

The results were as predicted.

*Playing Favorites*

In Example 1, what is the probability that the student takes a lemon square? An apple? A slice of apple pie? Express your answer in terms of percentages.

The probability that a student takes a lemon square is 25% which is the same probability for taking an apple and a slice of apple pie.

In Example 2, for what values of randNum would the student take a lemon square? An apple? A slice of apple pie? What does this code do when randNum has the value 9?

A student would take a lemon square if randNum is 0, 1, 2, or 3, the student would take an apple if the randNum values are 4, 5, or 6, and the student would take a slice of apple pie if the values are 7 or 8. If the randNum value is 9 then the student only takes the initial sandwich and coffee and then sits down to eat.

Overall, what behavior does Example 2 in the table above simulate? Express your answer in terms of percentages.

Example 2 expresses a 10% chance of the student only taking a sandwich and coffee and the other 90% will take the sandwich, coffee, and one other item which will have a 40% chance of being a lemon square, a 30% chance being an apple, and a 20% chance of being a slice of apple pie.

How would the behavior of the code in Example 1 be different if the first two else keywords were left out; in other words, if it were a sequence of separate, independent conditional statements? Could you write code that would have the same behavior as Example 1 without any else keywords?

This code would work because the if statements in example 1 use the == statement so separate if statements would still call the correct action. The code could also be written without any else keywords if the final statement is changed to if (randNum == 3).

How would the behavior of the code in Example 2 be different if both else keywords were left out; in other words, if it were a sequence of separate, independent conditional statements? Could you write code that would have the same behavior as Example 2 without any else keywords?

If the two else statements were changed to independent, conditional statements, the code would do more actions than intended because statements after the first if would execute the code if the integer is less than 7 or less than 9 no matter if a lemon square was already taken. To write code for Example 2 without any else statements, we would have to use the “and” operator by having the code look the following: if (randNum > 3 && randNum < 7). This would only execute the action if the randNum has a value equal to 4, 5, or 6.

How would the behavior of the code in Example 1 be different if it repeatedly called the nextInt method and checked its return value against different values instead of calling nextInt once, saving the random number in a variable, and checking the variable against different values? In other words, if there were no randNum variable, and each conditional expression called the nextInt method instead?

If the code was set up in this way then the student would still get the sandwich and coffee, but it would be possible for the student to get one or more food items outside of that or no more at all. This is because by having each conditional expression call its own random integer, the decision to execute a single action does not impact the decision for the other actions and each are evaluated separately.

*Showing an Independent Spirit*

Where should you add your complex expression?

The complex expression should be added in the main method where the if statement is created for changing direction of a fish if it is at the wall. This program can be edited to include random direction changes.

Do you need to construct a new Random generator object for this exercise, or is there a generator object that is accessible to you already?

There is no need for constructing a new Random generator object because there was one created earlier in the main method. However, the generator.nextInt(4) action needs to be called in order to get new random integers.

Should you use an AND (&&) or an OR (||) operator in your complex condition?

The OR (||) operator should be used because the fish can randomly choose to change directions even if it is not at the wall or the fish may be at a wall but the other half of the condition is false and the fish would still need to change direction. One or the other has to be true to change direction, but both do not need to be.

Does it matter which expression comes first in your complex condition?

Order does not matter for the complex condition because the if statement only needs to see that one is true and the code will then be executed and it doesn’t impact the results if the program analyzes the condition in different orders.