Transcript –

Software Design and Development

(Duration 24 minutes 53 seconds)

(upbeat music)

[Please download the handouts and have them available while watching this video]

Welcome to the HSC Hub. As part of the HSC on demand initiative, Secondary education staff have created a series of 45 minutes support modules for HSC teachers to use with their students. I'd like to begin by acknowledging the traditional owners of the land on which we meet today. I would also like to pay my respects to Elders past, present and emerging. Aboriginal and Torres Strait Islander peoples are advised that this video may contain images, voices and names of people who have passed away.

[Algorithms and desk checking]

The software design course includes key concepts, knowledge and skills around the production and checking of algorithms. These concepts, knowledge and skills around the logical production and checking of algorithms. We'll be looking at two areas in this presentation. The first is iterations, conditions and operators. Including how these things affect each other, and types of loops and how the same logic can be applied with these loops. I'll start this section looking at a question. This is a multiple choice question from the 2019 HSC. Only one third of last year's cohort were able to answer it correctly.

[refer to SDD HSC 2019 Question 10]

The most common incorrect answer was actually C. When approaching a question in the HSC, you need to determine what knowledge is required. So let's unpack this question so we can determine what this question is actually asking about.

The first thing is it's about algorithms. It states that in the question. The NESA course specifications document on page 43 shows these two fragments of the algorithm as types of loops. The first is a pre-test loop. And this is shown in the parts of the questions that are A to D. A post-test loop is shown actually in the question stem. As part of this question, you need to understand the difference between those two types of loops, and what the test or condition is. The processing will continue in a loop or will stop in the loop depending on whether the condition is true. In a pre-test loop, the processing condition whilst ever the condition is true. It'll keep going in that loop whilst ever that condition is true. A post-test loop is different in that the processing will stop once the condition becomes true or is true. So, they are different loops that can perform the same logic.

So, a condition compares things. So, in that loop, the condition is comparing two things. There's a good little example there. What we're saying here is that a statement that X is less than 10. In this case, X is a variable. And that statement X is less than 10 will be either true or false. For the condition to be false, X must be either equal to 10 or greater than 10. For the condition to be true, X must be less than 10.

If we're trying to find out which of the pre-test loops is the same as that post-test loop in the question, we need to focus on the condition or conditions that those loops have. So this is a simple example. [refer to Activity 1 from the student booklet] It's not the one from the question, but these two loops have the same logic. They are different types of loop but perform the same thing.

Notice firstly that the condition has moved in the REPEAT loop to the end of the loop. And also, that the relational operator, that is less than and greater than or equal to 10 has swapped. The left WHILE loop continues to dothisstuff, as long as count is less than 10. The REPEAT loop stops repeating dothisstuff when the variable count becomes greater than or equal to 10. By the way, don't be too concerned about the processing that dothisstuff actually does. We don't know what dothisstuff does. We're just focusing on when it’s going to stop being done.

So let's go back and look at question 10 from that HSC. It has different sets of conditions. We call this one a complex condition because it's got the two conditions joined. It's the combination of two comparisons linked by either an OR or an AND. The AND and the OR are known as logical operators, as opposed to the other operators we saw before. In order to work out whether the entire condition is true or false, we need to evaluate both conditions on either side of the AND or the OR and to see if they form the overall condition as being true.

So let's look at the two conditions of the complex condition. In X is greater than 5, or Y is equal to 8, for the overall condition to be true, either X can be greater than 5 or Y is equal to 8. For example, if X is equal to 7, and Y is equal to 7, then the overall condition is true because one of them is true. In X is greater than 5 and Y is equal to 8 for the overall condition to be true both of X is greater than 5, and Y is equal to 8 need to be true. So, if X is equal to 7 and Y is equal to 7, the overall condition is false, because only one of them is true. [refer to Activity 2 from the student booklet ] The answer to question 10 in the 2019 HSC, is actually B. whilst most students got it wrong, they said C.

So let's have a look at what the students who got it correct are most likely to do to get the answer of B. The first thing they're going to do is understand that to change from a post-test loop to a pre-test loop, you need to reverse the relational operators in the condition. So both of them need to be changed. You're going to reverse the relational operator of greater than to less than or equal to. And the relational operator of equal to, to not equal to. That's not enough. You also need to reverse the logical condition that goes between those two from an OR to an AND. And that would come up with B.

[refer to Activity 3 from the student booklet ]

The other option that students might have used to get B was they guessed. Although that's not the approved method of finding the correct answer. So I suggest you have a go at doing these. So I've given this example here of WHILE A is less than or equal to 10 and B is not equal to 12, you can pause the presentation here and have a go at converting this WHILE loop to post-test or REPEAT loop. Get your teacher to check it once you've done.

[insert refer to Activity 4 from the student booklet ]

The second part of this presentation is looking at how we can check algorithm logic. Let's consider this as desk checking. So what is a desk check? Desk checking is a method of going through an algorithm to understand what it does and/or find out any logic errors that may have occurred during that processing. Whenever data input or a Get statement is occurring, you need some test data to test it. The desk check displays how the variables and the data structures within that algorithm, how they change throughout the algorithm's processing. And then we record them in a table so that we can see how they're changing. The desk check result table shows any output the algorithm produces.

The first piece of advice about deck checking is don't make assumptions. Students will often go through and think they know what's going to happen next. But what they need to do is process every line of that pseudo-code or algorithm to do exactly what it states.

So the basic step to these. First thing is find and list the variables in that algorithm. Draw a table using those variables at the heads of each of the columns. This should include just the variables and any other identifiers. If you've got arrays include the elements of the array. You also need an output column to show any output from the algorithm. If there's a Get statement, you need to work out what test data is required.

Once you've got those things, you can work through the algorithm line by line, processing each line just like you're a computer. As I said, don't make any assumptions. Record all the changes to variables or other identifiers, and their contents in that table. This is a simple SWAP algorithm.[ refer to Activity 5 from the student booklet]

You'll see there we've got the REPEAT loop and we're going to go through and look at what the variables are. So the variables in this algorithm are X, Y, and temp. There is that loop. So we're going to test that the condition only stops the loop when X is equal to Y. The test data I've chosen to test this loop are 1, 2, 3, 4, and then I've got two 6's. If you see this written in the exam, you use each value of the test data only once when the Get occurs. I've chosen the two 6's to test that condition and when X is equal to Y.

So let's have a look at working through this with the desk check table. Okay, as I said, we're going to record X, Y. We've got temp. And then, because we've got that display, we're going to record the output. So my first piece of test data was 1. So I read that first and X is going to be equal to 1. Then, because I've used the 1, when I get to the Get Y, I'm going to use the piece of test data, that's the 2. Then, the first piece of processing there then says, temp is equal to X. That means to make temp equal to X, so, X is equal to 1, so now, temp is equal to 1. The next thing I'm going to do is make X equal to Y. Y is equal to 2. So X gets changed to a 2. The next line in the algorithm says, what make Y equal to temp, temp is equal to 1, so Y is now equal to 1. Then, I need to do the display line. So at the moment, X is equal to 2, Y is equal to 1. So I'm going to display X and Y. So that will be 2 and 1. I'm now at the end of the loop. The condition in that loop says that it's going to keep going through this loop until X is equal to Y. They're not equal to each other. So that part of the loop is finished.

But I go back up to getting the next two pieces of test data, I'm going to start the loop again. The next two pieces of data are 3, so that's the Get X, and the next one is 4. So when I get to Get Y, the 4 gets put into Y. What I should be doing now is working through all of the lines in the algorithm until I get to where both X and Y are the same. I'm going to skip forward a little bit to show you what that will look when it's finished.

On this table, the area marked red indicates that the SWAP occurred even when the values were the same. [Swap 6 with a 6] So working through that algorithm, the 6 and the 6 are the two pieces of input, but it gets to the point where it still swaps them, even though they're the same. So the second from last line there says, 6 and 6, that's where they were input. It swaps them again to 6 and 6, and then the output is 6 and 6. After that display, the loop will stop, because 6 is equal to 6. The areas marked green [2,1 4,3 6,6] in my desk check table are indicating where the output is inside the loop.

So if the display is inside the loop, you'll get multiple pieces of output. The dark lines in the table, that is after the X is equal to 2 and Y is equal to 1, indicate that the values have changed inside the loop. So that's each iteration of the loop shown with the table with the dark lines.

This is the past HSC question I wanted to look at. It's Question 24 a from the 2019 HSC. [refer to Activity 6 from the student booklet] The NESA documents indicate that the better responses show the correct incrementing of X in their desk check. They display the correct outputs of 4, 8 and 12 and the word ‘Done’ when it's finished. They use a correct layout in the table and use a single column for each variable or identifier. NESA has said that those students who needed improvement or they had a weaker response, they needed to improve in checking the correct outputs. They needed to show the processing of the variables in the loop and demonstrate the processing as well as the output.

Using the steps I showed you before, the variables we can see there are X, Y, and Z. Because there's a print statement, we need an output column. There is only one Get statement in this algorithm. So we only need one test value or one piece of test data. The HSC question gives you this piece of test data as the number 2. When you're doing an HSC question, you really need to ensure that you use the value or values that the question actually gives you.

So let's work through this example. I'm going to work through this example from beginning to end. So as I said, we've got X, Y and Z as the variables. We also need that output column. First line says that we need to set X equal to 1. The next line is the Get line. And so, that value of 2 is what we use as this piece of test data. Line 3 indicates that we're going to start that post-test loop. This could happen a number of times. Line 4 says, that we're going to change the value of Z to what's currently the value of Y. Line 5 tells us, we're going to enter a binary selection. So that binary selection, we need to check if X is less than 7. This is where you have to use your desk check table. So if it's a condition where you're checking a value, in this case, X is less than 7, we're going to go back to the table say, yes, X is less than 7, because it's equal to 1. So we're going to do the processing in that binary selection. In this case, it says make X equal to X plus 1. So at this case, X is equal to 1, we're going to set it to what's currently in there, plus 1. So now X is equal to 2. We now start another binary selection.

Let's check to see whether X is less than 8. It's equal to 2, I can see that in the table. So then we're going to print X times Z. So X is equal to 2, Z is equal to 2. 2 times 2, the output will be 4. The two binary selections are then finished in lines 9 and 10. And in line 11, we're going to make X equal to X plus 1 again. 2 plus 1 is 3. We're at the end of the loop. Is X greater than 8? This is the condition for the end of the loop. It's not, we can see the 3 is not greater than 8. So go back up to the top of the loop. And we we make Z equal to Y. Y is still equal to 2. Whilst Z is already 2, because that line of processing is there, I'm going to write that as being made equal to what's currently in Y. 'Cause we're checking what's in Y and making Z equal to Y.

The binary selection again, is X less than 7? Yes, it is less than 7. So X equals to X plus 1. The next binary selection is X less than 8? Yes it is, it's 4, so it's less than 8. So then, we're going to print X times Z. 4 times 2 is 8. Lines 9 and 10 end those binary selections. So, line 11 says make X equal to X plus 1. Finish the loop again. Is X greater than 8? No it's not. So we’re going to go back up to the top of the loop and make Z equal to Y. Y is equal to 2, we're still making it equal to 2. Line 5 again, has the binary selection is X less than 7? So 5 is less than 7, so we’re going to make X equal to X plus 1. Another binary selection. So we're going, if X is less than 8. It's still less than 8. So we’re going to print X times Z. 6 times 2 is 12. The two binary selections, finish again, and they then make X go up by 1 and the loop finishes. X is still less than or equal to 8. It's not greater than 8 yet. So Z is equal to Y again. So this time when we get to line 5, it says if X is less than 7. It's now equal to 7, so it's not less than 7.

So we’re going to skip that binary selection. That means we’re going to jump all the way down to line 11 again. All of the processing outside binary selection doesn't get done. That's why we skip to line 11. At this stage, we make X equals to X plus 1. So it's equal to 8. The printing of X times Z doesn't happen at this point. That would only occur whilst ever X is less than 7. We do go to back up to the top of the loop because X is only equal to 8 and not greater than 8 yet. We go, the loop finished equal to 2, because we're making Z equal to Y. We make X equal to X plus 1. So we end up with 9. This time, X is greater than 8. And so we get to line 13 and we print Done.

This is one I worked through earlier similar to the one I just had on the board. You can see there that each iteration of the loop is indicated by the ruling off the line. And the loop finished when X was greater than 8. This is what NESA says about this. So it's important that you always look at the marking guidelines when you're looking at past HSC questions. The column headings are clear in this example that include both the variables and the output. The first values for each variable are clearly shown. Each time a value is set or reset, it is shown in the table. Even though that Z is being set back to 2 each time, you still put that in the table. All outputs are shown, including when they are shown in relation to the values on the variables, particularly X. These are just the acknowledgments of where I got the resources for this presentation.

So that was just a brief run through some past HSC questions, just to show you how you can work through those questions on algorithms. Remember that there's algorithms form a large part of any HSC in Software design.

(upbeat music)

End of transcript