

Transcript for - “From Psychology to Logic: Learning Computer Programming in the Kitchen”

[MUSIC PLAYING] PROFESSOR ALI: Hi. I'm Tanzeela Ali from Superior University, Pakistan, majoring in Computer Science, [? still, ?] and my favorite subject is Programming, especially when it comes to building the logic for a program, or relating the real life stuff with programming one.

And in today's lesson, we're going to do the same thing. We are going to learn programming right here, working in the kitchen. We will analyze that, where are those basic programming concepts are applied while making a mango milkshake. So help me making the milkshake, and I'll help you learn programming. Deal? So let's get started.

While programming, there might be some situation when a certain task is assigned to you. Well, you know how to do this by using the marvelous brain you were blessed with, almost without thinking. But when it comes to parting the task into discrete logic, then you get stuck. Why does this happen?

This is because the human thought process is much faster than the sense of observation. Life experience has enabled us to lump together a set of steps into one macrostep. Driving a car is an example, as it has become so much intuitive, you don't need to think when to apply the brakes, for instance.

But the computer does not have the advantage of life knowledge. It's a dumb machine, even it can't think anything. So being a programmer, it's your challenge to move your smarts all your knowledge into discrete sequenced steps that will direct the computer what to do, also called a program. So if this thinking process could be slowed down, you would be able to identify the steps that were followed by your brain, and eventually, you will make logic for a computer program.

Moving towards my mango milkshake, as a first step of recipe, I have to mix sugar with milk by using this blender, but I have poured the ice already in it. Now, it's not a good approach to mix sugar in cold milk, so what I want is, I want this ice to be in this bowl, and this milk to be in this blender. But how can I do this?

Think about this problem. Discuss with your fellows and teacher. And I'll see you in a while.

Welcome back. I hope you had a healthy discussion. Well, it's good if you have come up with something. Well, what's the solution? If I had to pour the milk in the blender, then it needs to be emptied first. So to empty this blender, I need some container to hold the ice for some time.

So I can use a plate. So all the ice from this blender will have been shifted to this plate. Now the blender's empty, so all the milk can be converted into the blender. Now the ice will go back to the bowl. So you have seen, this way, we have swapped these two containers, the ice in this bowl now, and the milk in this blender.

Or we can also say we have swapped two variables. Hm. But how it is related to programming? Let's make it more specific.

What if I take a bowl of water and place it in front of me? So if I have swap the contents of these two bowls, what I have to do is, I need another container to hold one of the items. Because I can't hold the water in a bowl having ice already.

So what I need is, I can take another bowl-- here's all the ice in this bowl. The water will be shifted to that one. And the ice will go back in that bowl.

The same rule is applied on variables. When you have to swap two variables, you have to take another-- or you can say, a temporary variable, that can the value of one just for a short time. So if I call this one A and this is B, and that bowl was T, the code will be, T will be assigned the value of A, A will be assigned the value of B, B will be assigned the value of T.

So you can see how we have dragged the logic for a program out of a routine life task. Now, there are so many examples out there. You just have to explore them.

Well, moving towards my recipe, time to add a mango in the milkshake. Now, I like mangoes so much, and I want the biggest mango to be added in my milkshake. But I have a basket full of mangoes. How can I find out which one is the biggest one?

If I have to compare the two, I can find out which one is the bigger, but how can I do this basket full of mangoes? Think about this problem. Discuss with your fellows and teacher, and come up with your findings. I'm waiting for you.

Welcome back. I hope you have come up with something. Well, let's head towards a solution.

If I have to find the biggest out of this basket, what can I do is, I can assume that this mango, randomly, is the biggest one. But it's necessary for my assumption to be true. I may be right, I may be wrong. So what I did is, I need to compare this mango with all of the mangoes in the basket. So let's observe what we are going to do.

I assume this mango is the biggest one. So I'll pick any mango from this basket. OK, this one. Let's compare those. To compare these two mangoes, I have to find out the mass of these mangoes, so let's calculate.

This mango has a weight of 260 grams. OK, this mango has a weight of 448 grams. So it shows that this mango is the bigger one. I'll put this mango down, outside the basket, and compare it with the next one.

OK, let's pick this one. It has a mass of 468 grams. So this is the lighter one, which means my assumption was not right. Now, this one is the biggest. So, I'll put this down, and keep computing it with all the mangoes, until the basket is empty.

So I have this mango in my hand of 468 grams. But I can't tell whether it is the biggest one, because there are two more left in the basket. So I'll take another one. Let's calculate the mass of it. It has a mass of 265 grams.

So it is not the bigger one, so I put it down in the plate. Let's take the last one. It has a mass of 675 grams. Now, 675 is greater than 548, so this one is the bigger one, so I

can say it is not only the bigger, it is the biggest mango. This way, we have find out which is the biggest mango out of this basket, but how can we write the code for this? Let's form it into an algorithm.

Assume the first one is the biggest. Compare it with the next one. If the next one is smaller, put it outside the basket and go to Step Three, else, if this is the bigger, consider the new mango to be the biggest one. Repeat Step Two until the basket is empty.

OK, done with the algorithm. Now, let's see how we can write the code for this. First of all, we have to see how much memory, or I should say, variables do I need? We used a basket to hold some mangoes of different mass. For our program, we need a set to variables of same type, but different values, and this is what an array is made for.

An integer array can be used to store the masses of mangoes, and let's name it mango. Also, I used my hand to hold my assumed biggest mango, which is a single memory location. So we need a variable. Hm. Its name it max, of same type as the mango array is. Now, let's write the code.

I assume the first one is biggest. Here, I'm calling the first index 0, because array indexing starts from 0. This is our initial condition. Now I have to check it with the next one. Let's call the value of the current index i . If the condition is satisfied, max will be replaced by the current mango. If not, then there is no need to do anything. So we're not going to write the code for else.

Here, we have repeated the task of comparison, so we need a loop to do that. The loop has to repeat as many times as the number of mangoes in the basket minus 1. There are five mangoes. The code for a loop will be, for i equals 1 to 4.

Congratulations. We have written the code to find the maximum amount of an array, and a biggest mango for my milkshake as well. Now, time to cut it, but where is my knife?

I don't know where did I place it, and I can't find out in this messy arrangement. Well, I've got an idea. Well, look at these jars. They need to be arranged according to their heights. What if I have to arrange these jars?

Um, if you have to arrange these jars, what step do you take? You might have done this many times in your life, but today, you have to think in a step-by-step fashion. Take a few minutes, and I'll wait for you.

Welcome back. I hope you had a healthy discussion. I've cleaned my kitchen as well, and found my knife under the plate. Well, now it's time to arrange the jars.

These are the jars in not a good order. Let's arrange them. Let's align them in a line so we can compare it better. What we're going to do here is, we are going to apply the previous knowledge we have learned to make a new algorithm. One approach that we can follow is, find the biggest out of these five, and place it on the first place.

So let's see who is the biggest one. We don't need a scale at this time, because we can easily calculate which has the tallest height. So this one is the bigger out of these five, so I'll put this in my hand. It will be replaced by the jar on first place.

So the first will go here, and this jar, which was on the third place-- well, we don't need to care about the previous position-- we'll place on the first place. OK, now what we have to do next is, we will find the biggest out of these four, ignoring this one, or we can say, the second biggest. So we can see that this one is the second biggest, put this in your hand, and swap it with the one on second place. Done with that.

Now, can you guess what is the next step? Well, if you have come up with something, that's great. Well, the next step is, we have to find out the third biggest out of these five, which is this one, and we'll replace it with the one on third place. Now, there are these two left, and we have to find the bigger out of these two.

This one is the biggest thing, you can see easily. So this will be replaced by the jar on fourth place. And about the last one, there's no need to do anything, because it's already on its place. So you can see, this way, we have sorted these jars. Let's code.

We need an array to store the height of each jar, so we will take an array of size 5 with the name, jar. These were the basic steps that we have to follow in order to sort those jars. Find out the biggest jar, swap it with the jar on the first place. Repeat until the end of array. So let's code.

First, find out the biggest jar. I'm going to copy the code here that we have learned in the previous segment, with a little modification in the code, because we need to know the location of the index of the element of array. So the code will be like this. Secondly, we have to swap the biggest index with the first one. So here it goes.

Up until now, we have written the code to find the biggest and place it on the first place. Now, we have to repeat it one lesser than the size of array. Here, a repetition is needed, so we will use a loop here. A for loop would be perfect, starting from 0, and one less than the last index. Now the code would become like this.

One little modification is needed as well. For the first time, I needed the biggest one to be swapped with the first element, then the same action will be done for the second, and so on. So the index in the swapping will become like this.

Also, for the first time, I needed the biggest out of five, which was then swapped by first element. Then, I needed to find the second biggest, which can be called the biggest jar, ignoring the first one. Third time, when I needed to find the third biggest jar, I ignored the first two. This means I will start from finding the biggest jar, from the third, so here, a little modification goes. Instead of starting with 0 every time, the inner loop will start from the value of the outer loop variable.

We have done with the coding. Guess what will happen when we run this code? The whole array will be sorted. Well, this technique of sorting is called a selection sort, in which you select the biggest, then place it on the first place, and so on. Well, there are so many other ways to sort the array. You must give them a try.

Hey, why don't you give them a try. While I'm cutting this mango, go and think about other ways in which you can sort the array using another algorithm. So discuss with your fellows and teachers, and I'll see you in a few minutes.

Welcome back. I hope you have find out another algorithm to sort the jars. Let's discuss what's in my mind.

To sort these five jars, the technique I can use is, I can take this jar and can see whether it is the bigger than this one. If it is bigger than this one, it's OK, because I want the bigger on this side. So it's bigger than this one. It's all right.

Is this jar bigger than this? No. There Is a need to swap these two. OK, now is this jar bigger than this one? Yes. There's no need to do anything.

OK, now compare it with this one. Is this smaller than this one? Yes. So there is a need to swap these two.

This is one pass only, but we have to repeat this process until the whole array is sorted. Now, let's compare it again. Is this jar on its right place? Yes. It's bigger than this one.

Is this jar? Yes, it's bigger than this one. Is this jar bigger than this one? No, it is smaller, so we have to swap these two. so here it goes.

And now you can see that all these five jars are arranged. This technique of sorting is called bubble sort, which is also a technique that can be used in order to sort an array. Well, I'll leave the coding part up to you. Now, it's time to make the milkshake, and it's getting too much late.

OK. I've got the mango all ready in the break. So it's time to put all the ingredients. Now it's time to finalize my recipe. The blender has milk with sugar already in it. Time to put mango in it.

Now, blender has all the ingredients in it. I'm done with my desk. Now, it's just time to give a call to this blender, so the rest of the processing will be handled by it. Well, do you find any similarity?

Blender takes ingredients as input, and gives you milkshake as an output. It does some processing on it. In terms of programming, what would you call it? Have the debate on this, discuss with your fellows and teachers, and let's see what would you discover. I'm waiting for you.

Welcome back. I hope you had a healthy discussion, and it's great if you have come up with something. Let's get back to work.

While working in the kitchen, I use many machines to make my work easier for me. For example, if I have to make curry for the dinner, I don't need to chop the onion by myself, make the tomato paste by my hand. So I'll use machines to do this. I'll use my chopper to chop the onion, or use the blender to make the tomato paste.

The same concept is applied in programming. When you have to do a task, you don't need to start it from scratch. There may be some code already residing there. Well, a function is what is useful. A function is a useful piece of code that is written separately, and can be called when needed.

So remember the example of sorting the arrays in which we copied the code of finding the max? This is not a good approach though. What we can do is, we can make a function and name it, findMax, so all the coding of finding the max will go in that segment.

So if I talk about the input of my blender, well, it takes the ingredients of milkshake from me, and gives me milkshake as an output. So think of a blender that don't ask for the ingredients and always gives you a mango milkshake. Well, that would be all right if you want a mango one all the time. But what if you want a strawberry milkshake, or a banana one?

So you need something that can direct the function that which is required in order to proceed further. All these are called parameters, or arguments of function. Coming back to my findMax function, it would take an array as input and find the max out of it.

We also need to tell it where to start, and where to end the array. So two parameters for first and last index. As a whole, there are two parameters now.

Talking about the output of my blender, think of a blender that takes ingredients, process it, but don't give you any output. Will it be of any use? Absolutely not, because all the processing and ingredients will go waste. So a function needs to return something. Now, this is what a return type is used for.

In case of my findMax, it has to return the index of array where biggest value resides. So the code will become like this. Now, I just need to call the function in the sorting code, so the code will look like this.

Done with the functions. And we have done with the ingredients as well now. It's just time to make a call to it. So let me press the button, or in programming, let me say, I'm going to call this function.

OK, I think that's enough. Hm. My milkshake seems to be ready, I guess. Yes, it is. Well, if you were making a milkshake as well, enjoy, but if not, you have a more interesting thing to enjoy. That's programming.

It's a whole new world to explore, where you can show your creativity. And whenever you get stuck, remember the example of this kitchen, where we were making a mango milkshake, and we learned how to program something. And at the end, I would just like to thank you for watching my video and giving your precious time to me. Enjoy!

[MUSIC PLAYING]

[MUSIC PLAYING]

Hi. Welcome to the Teacher's Guide. Most of all, I would like to thank you for giving time to my module. This module is design for the students who are having difficulty,

while showing creativity in programming as they get stuck when they have to build the logic for a program.

Most the time, the solution is going on in the brain, but they're unable to convert it into a computer program, as they can not observe the working of really fast brain which we are blessed with. To overcome this problem, one must slow down the thinking process, and observe what steps were followed by the brain in order to solve a problem. And eventually, they will be able to identify the solution for a computer program.

This lesson is designed for a 10th-grade student who has a basic introduction with programming concepts that for, if – else arrays, in any language that may be GW-Basic, or C. This lesson would help them analyze that the real-world problems are closely-related with programming. And by watching this module, they may be able to make their own solution applicable for a computer program.

Well, the course of programming is pretty much different from other science courses, because the students are expected to show their creativity at the very beginner level. What they need is proper guidance about this course and, you can say, a good start for this. Now, this lesson is divided into six segments with five breaks, and each break has an activity for it.

Like, for the first break, you may ask two students to stand up. You may ask a student to hold two boxes in her hand, or his hand, and make them realize that if they have to swap the boxes in their hands, they need someone else to help him. So eventually, they will come to the point that, to swap the variables, you need another variable to hold one value temporarily.

In the second activity, you may ask the students to divide into a group of five. And so you may organize a competition that, who is the tallest one? In the third activity, you may ask them to organize themselves according to their heights. Emphasize on the process that they follow, or you can ask them to write it on a notebook.

Well, the fourth activity is just like it, but you have to emphasize that you have to find another way to sort yourselves. The last activity focuses on the modularization of problem. You ask them to divide the students into a group of four, where one is

the leader and the other three knows mathematics. One knows addition, the other one knows multiplication, and the third knows division.

And if the leader has to solve a problem, for example, adding two numbers, then multiplying them, then dividing them. So the leader does not has to know the mathematics behind this, but he can just call the fellows to do his work. In this way, you may make them realize that, where is the need of function in the programming.

I also encourage my students to speak as much as they can, because the more they speak, the more critically they will start thinking. And when this thinking process becomes critical, solution to a problem becomes clearer. The lesson summarizes all my findings and successful experiments done on my students, and I really hope it would be useful for you as well.

At the end, I would like to thank you again for giving your time and consideration to my module. Thank you so much.