

Pīkahu Name: What is Programming?

Video Name: Skills involved in writing a program (EMP07-4)

Presenter: Professor Tim Bell

Now that we have had a look at a program being written, let's have a look at the process that's involved. There are five steps that we can identify - it doesn't always happen exactly like this, but I think this will capture the main things that are worth knowing about.

The first thing is **analysis**, we need to analyze what the program is going to do, and for that we need good **communication skills** because we need to think about the person who is going to use it; or even meet with them, talk with them, the stakeholders. How is this program going to work? What's it going to do for your situation? I chose a very simple program and I just made up that we wanted something that you could type in a number and it would display it for you in te reo, but ideally I would go and find someone who needed it and find out what they really want. So I need good communication skills for that.

Then there is the **design** of the program - how am I actually going to turn that into something that the computer can do - and that requires a bit of **creativity**. I had to think about, well, maybe I'll just say 'Is it one?' and then display tahi, 'Is it two?' and then keep displaying each thing, but there are other ways I could have done it. I could use some more advanced tools that would have made it a little bit simpler, but that's the design part of it.

Now that I know roughly how it's going to work, I have the algorithm essentially in my mind, or maybe I have sketched it out somehow; then comes the **coding**. The coding is just simply a **skill** where I go, 'Oh yes I want to test this and test that' and what is the instruction in the language that I am using for doing that? So I need to know the language, I need to know how to convert my idea, my design into the actual language.

Once I had actually done some coding or started doing the coding, I did some **testing**. I really wanted to make sure that it was going to work properly, and that requires **rigour**. It requires me to go 'Oh I wonder if it works for big numbers and small numbers? I wonder if it works for the boundary between nine and ten and eleven?' Those were the places where things might go wrong. Now normally you can't test it for every possible input - for my program I could. I could test it for every number from one to nineteen, I could also test it for things outside that range. What if someone typed in zero, what if they type in twenty and so on? But testing is a really important part of programming because inevitably there will be something wrong somewhere, even if you've just made a typing mistake.

After that, of course, once you test it and you find something wrong you have to **debug** it to figure out exactly what's causing it to go wrong. Sometimes that's really simple, sometimes it's quite challenging, and we're looking separately at how to teach debugging, but that requires **persistence**; that requires your students to really want to keep going until they find that bug. If they use the right techniques and if they have a few tools at their disposal to be able to track down bugs, it should generally be possible to do that within a reasonable amount of time, but you do have to be prepared to go hunting for those bugs and fix them rather than just give up and go 'I'll make do with how it is'.

Those are the elements or the processes that are involved, and as you can see, those skills there - **Communication, Creativity, Skill, Rigour, and Persistence** - they are all skills that we'd love to see students displaying outside of programming as well, so it is a great environment for getting them thinking about that. You can also see that they are pretty human centric skills. The coding, knowing the actual things to drag into place or the commands to type, is only a part of the whole business of programming.

So there's an overview of programming, again I just want to reiterate, I've jumped straight in and shown you a program. I wouldn't expect people who haven't done programming before to necessarily be able to make a lot of sense of it or to certainly be able to put a program together like that. That program is the kind that would be expected of a high school student, but it is intended to just illustrate the whole point of programming and the whole process of it. In the later Pīkau we are going to break this down and look at it a lot more carefully, at each of these elements.