Hello, I’m Neil Rickus and welcome to my first session on, “how to use tablets to teach programming in the primary classroom”. In addition to working as a primary school teacher in England, I hold a Computer Science degree, which means I'm fortunate to have a good understanding of the programming elements within the new computing curriculum in England. I’m also employed as a lecturer at University level, where I teach trainee teachers and I spend the remainder of my time working as an independent consultant, which includes running programming workshops for both pupils and teachers, and advising schools on how to implement programming across the curriculum. If this is of interest to you, or would like further information about my work, please visit my website at **computingchampions.co.uk** or contact me using the information on the screen.

So, during this session, we're going to be discussing tablets and specifically, the current situation in English primary schools with regards to computing, why we're discussing tablets rather than other technologies and then we'll begin examining a range of suitable apps.

Since September 2014, primary schools in England have been mandated to teach the new computing curriculum, which includes a number of elements related to computer programming. For example, from age 5 to 7 years, pupils are required to **sequence instructions** and **predict the behaviour of simple programs**, while from 7 to 11 years, pupils should know how to implement **repetition**, **selection** and **variables**. Unfortunately, when I go to work in schools, I often see teachers simply, “doing *Scratch*”, which is a popular, desktop based, visual programming environment, rather than focusing on the underlying concepts to really help pupils understand what it is they’re learning. However, by using tablets to teach programming, schools have a wide range of apps to both introduce and reinforce concepts, without having to be focused on a single piece of software.

So, we have the question, "Why tablets?". Firstly, schools are increasingly investing in tablets, such as the *Apple iPad*, or *Android* based devices, such as the *Samsung Galaxy Tab*, as they allow pupils to study in a wide range of locations and the devices generally, “just work”. An increasing number of schools are starting to get rid of the traditional computer room, which means they are limited in the devices on which pupils can learn to program. Despite the large amounts of money spent on tablets, schools are often struggling to provide evidence of their “Return on Investment” (or "ROI"), thus by also using the tablets to teach programming, management can provide further reasons to purchase devices.

As outlined in the table, which can be found on my website at **computingchampions.co.uk/mooc**, I've identified 17 *iPad* apps, which allow you to cover all the programming elements within the primary computing curriculum. Fortunately, the majority of apps are free. However, those apps suited to older children often include in app purchases, or require the user to sign up for an account, which also means the device requires access to the Internet. Some of the apps are also available for the *Android* operating system, although the choice is currently limited. Within the table, I've highlighted the suitable age range for each app based on the requirements of the computing curriculum and also whether they included a "creative mode", which allow pupils to produce their own content, rather than simply progressing through a set of increasingly difficult levels.

The four apps highlighted in yellow, *Daisy the Dinosaur, Scratch Jr, Hopscotch* and *Tynker,* provide a steady progression for pupils as they begin to learn more advanced concepts and require access to further functionality within their programs. During this session, I will demonstrate two of the apps for younger children, *Daisy the Dinosaur* and *Scratch Jr*, with the other two apps, *Hopscoth* and *Tynker*, in the next session. So, to the *iPad*….

Firstly, we're going to look at *Daisy the Dinosaur*, which is suitable for pupils when they first enter primary school at age five. As you can see from the title screen, pupils have the option of either, "free-play mode", where they have access to all the program's commands, or "challenge mode", where pupils have to undertake progressively more difficult problems. We'll have a look at "challenge mode". Within this first level, I have to get Daisy to move to the star by creating an **algorithm**, or a set of instructions to achieve a specific goal. Just like most visual programming environments, I have to drag the "move" command block into the area in the middle of the screen. I then tap "play" and Daisy wiggles to collect the star. Yes, I did it! On the second level, we now have to "move", then, "jump", with further levels getting progressively trickier. *Daisy the Dinosaur* allows pupils to **sequence instructions** and **debug** their programs, which pupils in England are already required to do at this young age, plus it also introduces **repetition** at a very basic level, which they have to master towards the end of primary school. The only negative aspect of the app is that is does require a basic level of literacy to read the command blocks, thus some prior work may need to be undertaken to help children understand the words. However, saying that, if you tell pupils they have to, "get Daisy to the star", they will usually figure it out!

The second app we're going to look at during this session is *Scratch Jr*, which is a simplified version of the popular, desktop only, programming environment, *Scratch*. What's great about *Scratch Jr* is, unlike the full blown version of *Scratch* and our previous app, is that all the commands are represented by images, thus eliminating the need to read. This also means it can be used by pupils speaking any language. Within the app, we have a number of built in sprites, or characters, so I'll choose the Purple sprite to add to my cat, along with a background, such as the beach image I'm selecting here. As with *Daisy the Dinosaur*, I can **sequence instructions**, such as getting the cat sprite to move towards Purple, which is executed, or starts, when I press the green flag at the top of the screen. I can also change the appearance of the sprite and get it to display some text, such as, "hi". I'll execute the program again, so you can see the results. *Scratch Jr* also allows you to use **repetition**, as outlined in the orange block here, plus we start being able to include actions based on specific events using, "if…then" type statements. For example, if I add in a movement instruction to make the cat touch the Purple sprite, I can add an instruction that says, "if cat bumps into Purple, then disappear", which you can now see if I press the green flag. In addition to helping teach the programming elements of the curriculum, *Scratch Jr* is often linked with Literacy lessons to allow pupils to produce short stories, or with other factual subjects, such as History, to provide an additional means to demonstrate learning.

So, that concludes our first session on how to use tablets to teach programming in the primary classroom. Our next session will focus on apps suitable for older primary children, in addition to how to incorporate these apps effectively into your lessons and record evidence of pupils' programming ability.