eping

The Lights On

*Virtual Engagement   
Lesson Guide:*

Code Like A Pro  
Software Development

****Introduction

**Lesson summary**In *Code Like A Pro* pupils act as software engineers working as part of a large team. This lesson introduces some of the basics of *Code Reviews,* and what makes *good* *quality code.*

Pupils are introduced to the idea of code quality, and the importance of writing readable code. They’ll get to comment on examples of good and bad code, and get hands on completing their own code review.

**Who is this for?**Anyone working in technology with a familiarity with the basics of software development. Volunteers do not need to be programmers, but a familiarity with computer code (Python) will be beneficial.

This lesson plan is an example of a high-quality software development activity suitable for high school age students studying National 5 Computing Science. It has been designed for remote delivery but is also suitable for an in person volunteer visit.

**How to use this lesson plan**1) Read through the lesson plan alongside the accompanying slides  
2) Customise the [slides](https://docs.google.com/presentation/d/1ME72K6omX7d-AhPEjDrG7I9IHiNQZ6PUSewx0WyegMs/edit?usp=sharing) to reflect what you do  
3) Look at the sample responses to some common questions (Page 11)  
4) Practice delivering the lesson, perhaps with a colleague or two  
5) Engage with the classroom!

Daniel Devine  
Digital Skills Education  
April 2022

# 

# Lesson Plan

|  | Code Like A Pro - Computing Science Volunteer Session  Volunteer:  Teacher:  Class Year/Level:  Sample Slides:  <https://docs.google.com/presentation/d/1ME72K6omX7d-AhPEjDrG7I9IHiNQZ6PUSewx0WyegMs/edit?usp=sharing> |
| --- | --- |
| Maximum 5 minutes | **Introduction**  Give a short, 2-3 slide, presentation on the area you work in and what you do. Think about what context the learners might need for the activity - in this case saying how you are involved in software development, eg. a programmer, software tester, analyst.  Make sure to include:   * Your name and how you’d like to be addressed * Where you are calling from * What your role is (a brief description of what you do) * Who you work for and what the company does     “Hello, I’m Eleanor Dee . Today you can call me Ellie. I’m joining you today from our oﬃce in Kinross. I am a software engineer at Big Grid Industries. I write code as part of a big team to solve problems other companies have.”    “Today you’re going to use your knowledge of software development to review and improve some code.”    “You’re also going to get the chance to ask me any questions about my role as a software engineer at Big Grid Industries.”  See slides 2-6 for inspiration. You may adapt these to match your role. |
| Main Activity Code Like A Pro | |
| 10 minutes | **Setting the scene**  “Today we’re going to talk to you about a really important step in creating software.”  “Every time a developer is finished with a piece of code for a feature, we do something called a code review.”  “What do you think a code review is? Have you heard of them before? Don’t worry if not, you can take a guess.  I’ll give you a clue - we’re not giving the code a rating out of 5 stars!”  “Type into the chat what you think we do in a code review.”  \*RESPOND\*  “At least one other developer will look at the code to see if they can understand it. It’s a type of peer review, I’m sure you do this sometimes - marking each other’s work.”  “We share our code with another developer and they will see if it does what it’s supposed to do, check for any obvious mistakes, and generally look at the quality of the code.”  **Reviewing someone else's code**  “Here’s what a code review might look like. Once some code has been written, we send it to someone else to look at.”  “They can leave a comment on any line by pressing the +, it’s a bit like Google Docs.”  **Looking at your reviewed code**  “When they’re done, the original developer can see all of the comments and changes requested by the reviewer.”  “Look here, we can see that Craig has said there are ‘some code quality issues you might want to address around variable naming’, and it even shows the parts of code where specific comments were made.”  “What I’d do next as the software developer is either reply for more information, or make the changes.”  “Then they get submitted again for review - hopefully this time it’s all positive comments!”  “The developer may not agree with everything that the reviewer has said. For example another reviewer may disagree on the use of emojis here, since emojis might not display correctly on the command line.”  **What are we looking for? What makes good code?**  “What do you think makes good code? What are we looking for in a code review? What makes good code?”  “Take 1 minute to choose your top 3.”  “Teacher name, could you time us for a minute?”  Okay great, tell us in the chat your top 3!”  \*RESPOND\*  **What does Google tell their developers to do?**  “Well this is what Google does in a code review. This is the real documentation that Google developers have to follow.”  “Just looking at this list really quickly, they talk about things like “*the code is well-designed, the code isn’t more complex than it needs to be, the developer used clear names for everything, comments are clear and useful*””  “Look at the paragraph at the bottom “*Make sure to review every line of code you’ve been asked to review, look at the context, make sure you’re improving code health, and complement developers on good things that they do”* It’s not an exam, it’s about helping each other write better code.”  “It’s not about finding bugs, or how fast it runs, it’s all about making readable code. Which is one of the most important skills you need to be a software developer.”  **Why does readability matter?**  “Good code is all about readability.”  “Here’s why it matters.”  “If code is readable, and easy to understand - we can find and fix bugs much more easily.”  “We work in teams, so if someone can’t understand it, they can’t contribute!”  “Good code will be around for a long time, so people in future need to be able to extend it, or add new features.”  “Imagine if the code wasn’t readable, we’d end up starting over and over again - that would be slow, and expensive.” |
| Maximum 5 minutes | **Looking at some code - JavaScript**  **What does this do?**  “Let’s look at an example of some code.”  “Have any of you ever used JavaScript before? Tell us in the chat!”  \*RESPOND\*  “If code is well written, we should be able to understand it, even if we don’t know the language.”  “What do you think this code does? Leave a message in the chat.”  “Do you think it’s good code? It has comments, it’s spaced out.”  \*RESPOND\*  “I’d say this code could be better, we don’t know what it’s converting to, do we? What is the data\_in? And what is the data\_out? We just don’t know the context.”  **What if it was more readable?**  “Here’s the code again, but with different variable names.”  “Tell me in the chat what the program does, is it what you expected?”  \*RESPOND\*  “Do you think you’d now be able to change it to convert kilometres to metres, or anything else?” |
| Maximum 5 minutes | **Looking at some code - Swift**  **What does this do?**  “Let’s look at one more example.”  “Have any of you ever heard of Swift? It’s what apps and programs for Apple devices are made in.”  “What do you think this code does? Leave a message in the chat.”  \*RESPOND\*  **What if it was more readable?**  “You can read this code out loud, it makes sense!”  “It’s the same code, but with better variable names, it’s just like English. Good code should be what we call self-documenting, we try to write code that is so clear, it doesn’t even need comments to explain what’s going on.”  “If you needed to change the welcome message, could you do it? What line would you edit?”  \*RESPOND\* |
| Maximum 10 minutes | **HANDS-ON CODE REVIEW**  **Guess a word**  “Let’s do a code review together.”  “We’ve got a small Python program that a developer has sent for code review. They’ve told us it works, and sent these rules along with the code.”   1. Read the gameplay description 2. Open Trinket and screen share <https://trinket.io/python/6b3aa9d269>    1. Screenshots (slides 18/19) are provided incase of access issue 3. Run the code, and play the game together to make sure it’s working   “Imagine this is a code review, and you’re the reviewer. In the chat tell us what comments you’d make. Tell us: the line number, and your comment”  \*DISCUSS THEIR SUGGESTIONS AND MAKE CHANGES\*  Some examples, these changes could improve the readability of the code by using more meaningful variable names:   * w -> mystery\_word * uw -> solved\_so\_far * counter -> guesses\_taken * sol -> solved   **What did we change?**  “Just with those changes, we made a big difference to the code. It’s so much more readable now.”   1. “We used meaningful variable names” 2. “We gave the code a clean and clear structure” 3. “We followed Python naming conventions and style (PEP8)” 4. “We added internal commentary when necessary”   **What have we covered?**  “We’ve gotten through a lot!”   1. “What is a Code Review” 2. “Readability is really important” 3. “You were able to understand well written code in different languages” 4. “You helped improve some code to make it more readable”   “Good job!”  “Working in software development is a really great career, hopefully you now know a bit more about what we do, and the skills you need to have!” |
| Questions and Answers | |
| At least 10 minutes | You should hold a Q&A for the remaining time of your engagement. Aim to hold 10 minutes for this.  If you’re running behind, skip an earlier section, not Q&A time.  By keeping Q&A till the end the pupils will understand more about what you do, and how data is secured in industry. This will assist them in asking deeper and more insightful questions.  We’ve gathered some frequently asked questions, along with sample answers on the next page. |

# Frequently Asked Questions

**Is being a software engineer a good job?**

“People working in software engineering on average get paid around £26,000 a year when they start. With 3-5 years experience, you can earn £35,000 to £50,000 [Prospects] a year. This is about the same as other types of technology jobs.

For example, at the time of writing (April 2022), CGI are looking for new technology graduates in the Edinburgh and Glasgow areas and will pay you a starting salary of £30,000. This is above the average salary in the UK overall (£29,600) [uk.jobted.com/salary] when you are just starting the job!”

**What school subjects should I take?**

“People in software development have a wide range of backgrounds. However, computing is probably the most important along with maths and physics. You should try to take these subjects as far as you can (Higher and Advanced Highers).”

**Where can I study software engineering?**

There are different options to study software engineering at many colleges and universities in Scotland. There are apprenticeships for school leavers, HNC and HND college courses, and BSc and MSc university degrees. Below are some examples:

* College
  + Edinburgh College – HND Software Development  
    Requires 3 Highers or above and that you are comfortable with a computer. No previous programming knowledge is required. This course grants access to the third year of a BSc at Heriot-Watt or Edinburgh Napier University on completion. (<https://www.edinburghcollege.ac.uk/courses/browse/software-development-hnd-cr1cosdb22>)
  + Forth Valley College - HNC Computing  
    Requires to be working towards 2 Highers, and 3 N5s. One of your subjects must be computing related.  
    (<https://www.forthvalley.ac.uk/courses/computing/hnc-computing-schools>)
* Apprenticeship
  + QA - Software and Web Development SCQF 8 Technical Apprenticeship  
    Anyone can apply, but you’ll need to demonstrate some knowledge and willingness to learn.  
    (<https://www.qa.com/course-catalogue/courses/software-and-web-development-scqf-6-8-apprenticeship-programme-scotland-qaadevl68s/>)
  + Glasgow Caledonian University - BSc (Hons) Graduate Apprenticeship  
    Requires 4 Highers at BBBB. Your time is blended between an employer, and the university.  
    (<https://www.gcu.ac.uk/study/courses/details/index.php/P03127/Graduate_Apprenticeship_Software_Development_for_Business?utm_medium=web&utm_campaign=courselisting>)
* University Undergraduate
  + Abertay University - BSc (Hons) Computing  
    Highers ABBB including Mathematics at B. (<https://www.abertay.ac.uk/course-search/undergraduate/computing/>)
  + Edinburgh Napier University - BSc (Hons) Software Engineering  
    Highers BBBB to include Maths or Physics. (<https://www.napier.ac.uk/courses/bengbeng-hons-software-engineering-undergraduate-fulltime>)
  + Heriot Watt University - BSc (Hons) Computer Science (Software Engineering)  
    Highers ABBB (including Mathematics). (<https://www.hw.ac.uk/uk/study/undergraduate/computer-science-software-engineering.htm>)

# 

# Next Steps

**Visit the Digital World Website**  
<https://www.digitalworld.net/software-engineer>  
The Digital World website will let you explore the different areas of software engineering available. You can find up to date information on where you can study, and other learning opportunities and resources.

**Play Cyber Skills Live activities**

<https://cyberskillslesson.com/>

Some good activities to start are:

[*Debugging Coding Disasters with Python*](https://cyberskillslesson.com/activity/debugging-coding-disasters/)- *In this interactive lesson, you'll use software development skills to fix a series of bugs in Python programs.*

[Cracking One Million Passwords with Python](https://cyberskillslesson.com/activity/cracking-one-million-passwords/) - *Look at a leaked file containing the login details of one million customers, then write code to try and work out the passwords on the leaked list.*

**We also run activities live!**Our interactive lessons don’t need any technical knowledge. By taking part, learners develop digital skills while learning about technology topics. Over 150,000 learners have already taken part live! You can register your interest here - <https://cyberskillslesson.com/>

# Acknowledgements

**Authors**

Daniel Devine

Craig Steele

[digitalskillseducation.com](https://digitalskillseducation.com/)

**Thanks to**

Anita Squirer

Kerri Knier

Matthew Jones

BJSS

Glasgow Gaelic School

Debbie McCutcheon