UCL

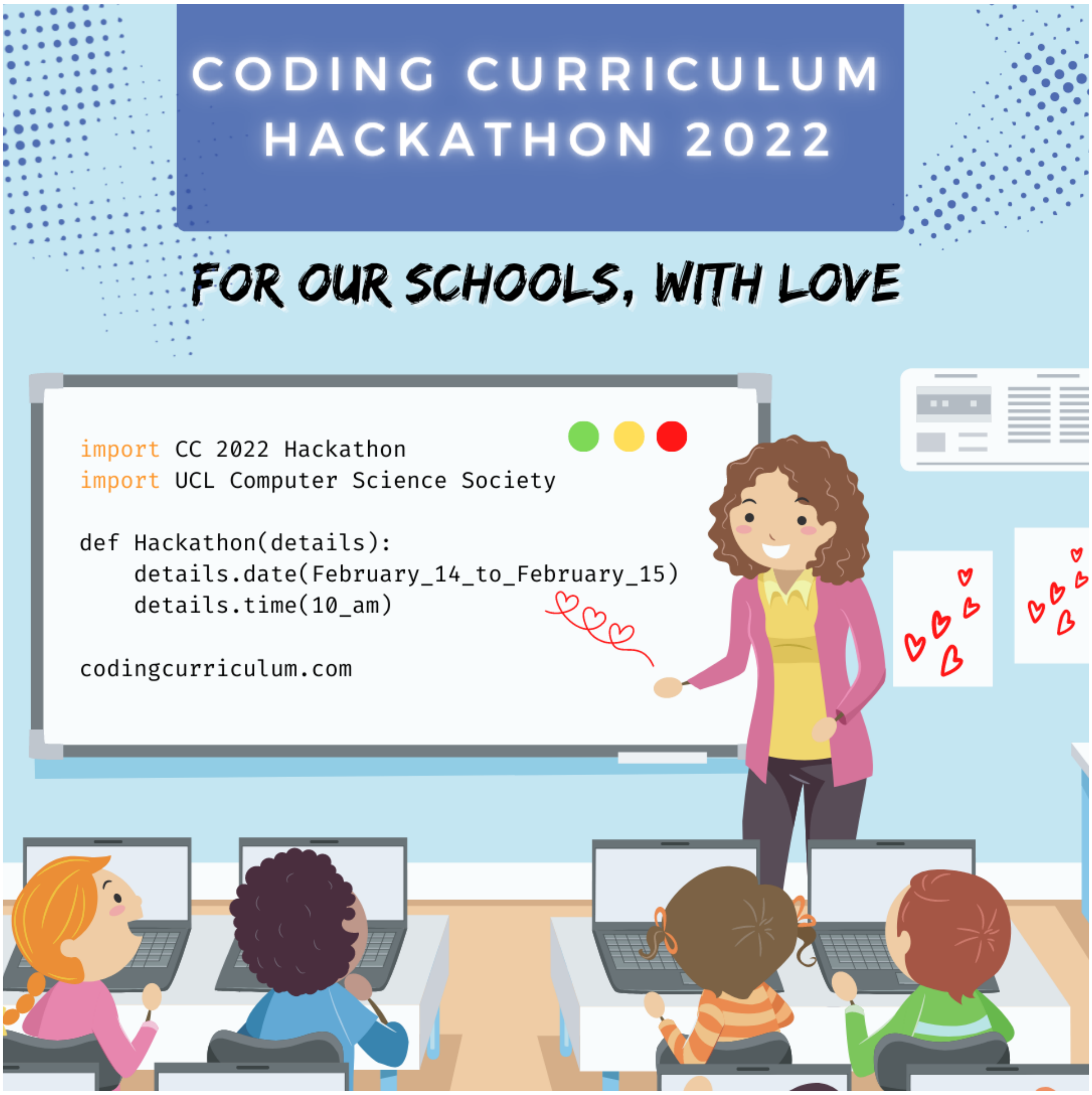
UCL Computer Science Coding Curriculum 2022 (CC-2022) Hackathon

For our schools, with love – A Valentines Week Hackathon 48-hour challenge

10am Monday 14th February to 9.59am Wednesday 16th February 2022

Submission of teaching materials: 10am Friday 11th March 2022

Codingcurriculum.com, and for UCL students: https://moodle.ucl.ac.uk/course/view.php?id=25742



```
import CC 2022 Hackathon
import UCL Computer Science Society

def Hackathon(details):
    details.date(February_14_to_February_15)
    details.time(10_am)

codingcurriculum.com
```

UCL Support: Prof Dean Mohamedally, Prof Graham Roberts, Sheena Visram, Rae Harbird, Vajeeha Farooq, Sam Stockdale, UCL Computer Science Society, UCL Engineering Society, CS EDI Group, CS Teaching and Learning Team, CS Strategic Alliances Team, UCL Institute for Healthcare Engineering, UCL Innovation and Enterprise

Invited Judging Panel: Microsoft (Prof Lee Stott), Microsoft Philanthropies (Sam Spilsbury), Alan Turing Institute, Amazon Web Services (Apeksha Jain & Courtney Waugh), InfoSys (Payal Chauhan), IBM (Prof John McNamara), NTTData (Tom Winstanley), Intel (Costas Stylianou), IXN for Good (Sheena Visram).

Open to all degree-level students, Teams of 3-4 maximum. The event can be run in your teams either remotely or based at UCL but will need to be checked with the Computer Science Society for registered on-site attendees (UCL students only on-site). Free snacks and drinks will be provided on-site.

Signup registration link - https://forms.office.com/r/9QnXs2jAq9

Venue: Remote for all students, or on-site available if you are a UCL student, at UCL Christopher Ingold building (14th-16th February).

Prizes

- First prize team - each member wins an Xbox Series S.
- The top 10 teams will win echo dots from Amazon UK.
- Certificates for the top 30 teams will be announced at the end of Term 2.
- Several companies HR and internships leads will be reviewing your submissions, including AWS, Microsoft and Intel
- Any students looking at further research please note the Alan Turing Institute for AI will also be judging in this event.

About

What if you could give your younger self an introduction to what today's state of the art in Computer Science can do? Is there something in your own experience, past or present, that you find really inspiring? What would you show you have learnt about - Machine Learning to make decisions, how Computer Vision can recognise you, or something else?

We are all grateful for the teachers and schools that helped to get us where we are today. Being able to clearly explain ideas to demonstrate what is possible in Computer Science can be incredibly inspiring and also a great validation of your learning accomplishments.

This Hackathon is a give back challenge to encourage new generations at your former schools across the world by enhancing teaching, ***using your understanding of Computer Science.***

Objective

This event celebrates your learning by asking you to work in teams of 3 or 4 to develop a ***Tutorial Package*** of ***extraordinary*** introductory teaching materials. You should aim to **excite** and **engage** young students and teachers at a school like the one you attended, giving them the educational materials that your team develops.

The best ***Tutorial Packages*** will become part of a growing open source compendium of education materials to be made accessible for all schools as part of schools outreach activities during the summer in collaboration with the Computer Science Society and various supported Schools Ambassadors programmes that UCL are engaging with.

Themes to choose from

The aim is to develop an interactive and exciting tutorial from one of these following themes, or suggest your own theme for something else that particular interests you:

1. Introducing what we can do with to AI and ML, including safe and ethical AI examples
2. Augmented reality on phones and tablets, 2D/3D graphics scripting with Unity or Unreal Engine
3. Apps development (mobile, cloud and web) with templates
4. Chatbots including Alexa and Bot Framework
5. Data visualisation, data processing and databases
6. Computer Vision including object recognition and gestures e.g. with OpenVino
7. Computer Games and interactive systems, including Minecraft
8. Cryptocurrency and bitcoin, financial computing concepts
9. Global and Local Health Informatics, Accessibility and Inclusion
10. IoT and sensor devices with Raspberry Pi or Arduino
11. Natural Language Processing and Speech
12. Networking models including standards, as well as social networks
13. Security and Encryption
14. Sustainability, charity and ecological development
15. Your own theme -> Send an email to Prof Dean Mohamedally for confirmation that your theme is suitable.

The design of the tutorial is only limited by your imagination and resources! You might, for example, look at data analysis using state of the art algorithms, explain how encryption works, demonstrate how image recognition works, it’s up to you.

You will need to create an attractive and interactive visual interface, which hides away complexities but still presents information and results in an easy to understand and school-friendly way. A walkthrough guide along with the demonstrator that you build should illustrate how to modify the tutorial and check that it works. This is part of a strategy we call **teaching by *scaffolding***, where teaching is guided by a set of carefully designed learning material complemented by interaction and experimentation.

Deliverables and Competition Rules

Tutorial packages can be developed and submitted in both English **and** in your own native languages.

You can make use of assets and code from your own learning and projects, current or past, but be sure to cite the sources properly. For easy sharing between your team members, your tutorial files and content can be stored in a GitHub repository that is kept private until the Hackathon is completed.

The submission of this hackathon work is in two parts:

1) A form and YouTube video is to be submitted by the end of the hackathon on 5pm Wednesday 16th February 2022 - https://forms.office.com/r/6zDaHD8AL0

   Fill in the submission Form link above, with the theme and project title. Include a 250 word abstract of what the tutorial is and how to set it up (text file). Include the YouTube link (no more than 6 minutes long) to a video of the tutorial package. **The judges will be keen to review these videos! To be submitted by 5pm on Wednesday 16th February 2022.**

2) All of the teaching and github materials will be submitted by **5pm Friday 11th March 2022** at the same link - https://forms.office.com/r/6zDaHD8AL0

   2.1 Create a runnable Build version (executable) for school students submitted in a zip file on Moodle (can be a cloud account download link via a text file). If it is in HTML it will open in a browser.

   2.2 Source code files with compilation instructions. It should have all of the necessary packaged libraries (open source) needed to run e.g., ML libraries, SQLite, GUI libraries needed for the solution.

   2.3 Tutorial guide for school students as a **Word docx** format document. This can be similar to the YouTube video. It should have walk through screenshots, in plain language aimed at a school's students. This should be in the style of an instructional blog post (and you can post it). Include versions of this in any localised languages that the team wish to, as well as English.

Note that all **documentation must be Word docx format** allowing it to be edited and republished. Other formats should not be used.

Your solution should not have any complex installation, or admin privileges required. You should not include services that need to be paid for by Schools. It would be **ideal if it runs as an executable, in a browser**, or at most just a few commands at a terminal.

Other advice: school children may have limited access to a Chromebook, tablet or Windows laptop, and likely a microphone, speakers and webcam. Browser or native GUI interfaces are recommended. The average specifications are currently to be 4GB RAM or less, with a Celeron processor as the lighter end of processor capability.

Example Tutorial Package (in a Word docx file)

Title: Enabling deaf users to see words as they are spoken using a speech to text engine

Team Authors First Names and Surnames:

Email addresses:

Summary:

(example) A template application for Visual Basic, Visual C# and Android to let you see words as they are spoken.

Keywords: Speech to text, Python, Android, Visual Studio

What you need to run this (ingredients): Windows PC with 4GB RAM, Python installed

Method:

students tutorial walkthrough with images, with installation and method to use.

Step 1...

Step 2...

Step 3...

Q & A: Any possible problems faced and signposting to additional materials and resources

If you wish to additionally publish your Tutorial Package online, a good example would be on GitHub following the Workshops style. This would be in addition to the word docx requirement – here is the Workshops style template:

https://github.com/microsoft/workshop-template/blob/main/workshop/README.md