ada computer science

Ada Computer Science Unlocked

New Web/SQL Projects, Class management, and more!

UNIVERSITY OF CAMBRIDGE

Raspberry Pi Foundation

What is Ada Computer Science?

- **Free** online platform built and maintained by the **Raspberry Pi Foundation** and the **University of Cambridge**
- Designed for **teachers** and **learners** (aged 14+)
- Written by computer science subject experts; based on extensive research and evidence about pedagogy
- **60,000+** registered users
- **5,600+** teachers
- **Over 4.3 million** question attempts
- **1,200+** self-marking questions
- **400+** real code examples

ada computer science

Raspberry Pi Foundation

UNIVERSITY OF CAMBRIDGE

Ada CS development

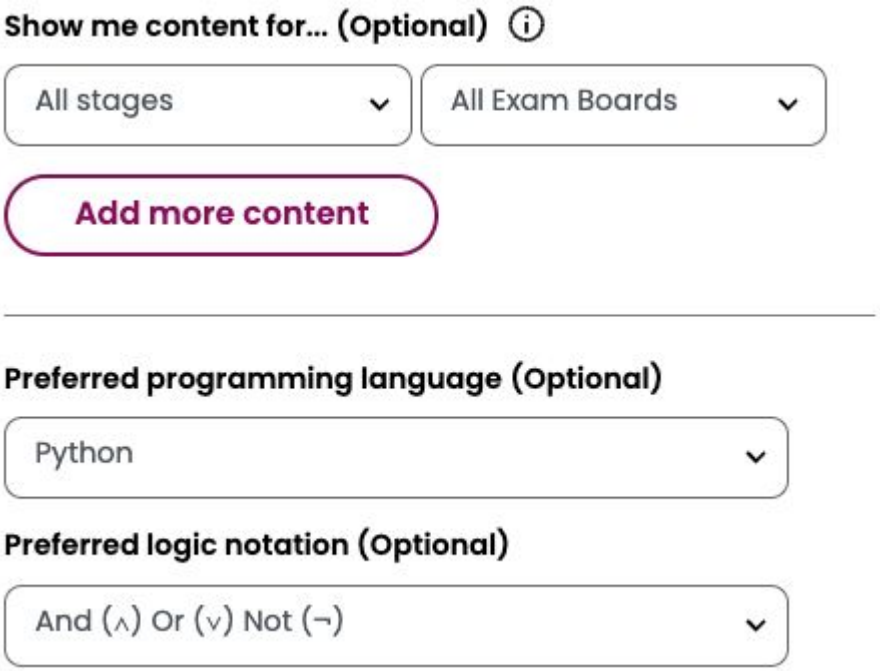
Since 2023, the teams at the Raspberry Pi Foundation and the University of Cambridge have developed the Ada Computer Science platform. We have:

- Written over **390 new questions**
- Introduced new question types including **LLM-marked** free text questions
- Introduced an embedded **SQL editor**
- Added exemplar code in Java and Visual Basic (VB)
- Reviewed and updated several topics including **new topics on AI and ML**
- Published **database, Python** and **web projects** to support with the NEA
- Improved the UI for teacher tools, new dashboards for teachers and students

13 strands of learning

- AI and machine learning
- Algorithms and data structures
- Computing systems
- Creating media
- Data and information
- Design and development
- Effective use of tools
- Impact of technology
- Models of computation
- Networks
- Programming
- Safety and security
- Software projects

Customised views for teachers and learners



All content can be filtered and organised by the user.

Concept pages

Systems architecture refers to the structure and organisation of a **computer system**. It specifies the **internal components** that make up a computer system and describes how these are **interconnected**, how they interact with each other, and how they are managed.

GCSE, A Level	**Common system architectures** ›
GCSE, A Level N5	**Internal components of a computer system** ›
GCSE, A Level N5	**The components of the processor** ›
GCSE, A Level Higher	**The fetch-decode-execute cycle** ›

Program code

Pseudocode | Python | C# | VB | Java

In Python, using the `range` function with three arguments allows you to define the **starting value**, the **stopping value**, and the **step value**.

Remember that the **stopping value** of the `range` function is **exclusive**. This means that the final value of `i` will be up to but **NOT** including the stopping value of 31.

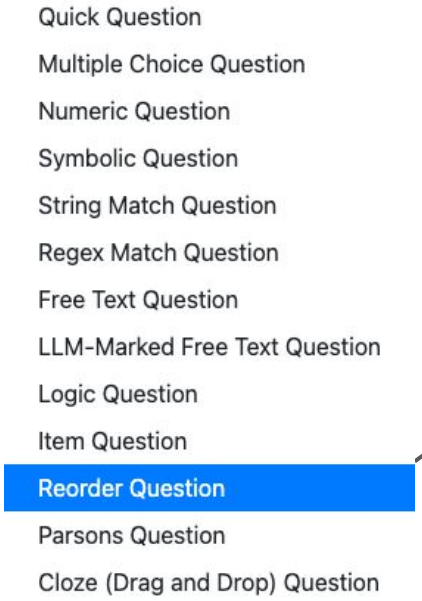
```
1 for i in range(3, 31, 3):
2     print(i)
```

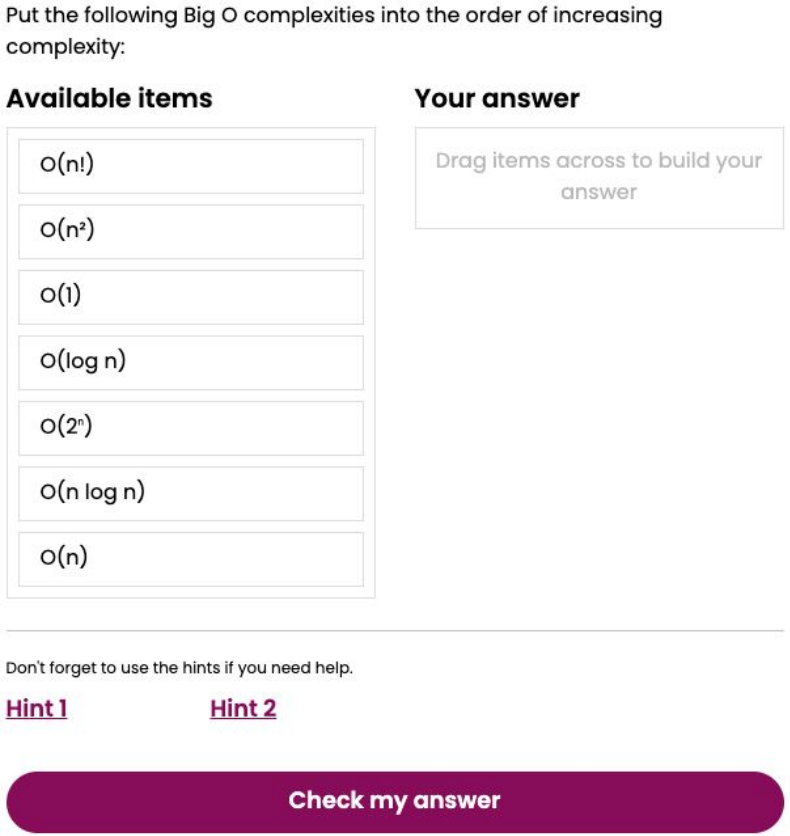
View on GitHub

Hundreds of downloadable, executable code examples, in multiple programming languages.

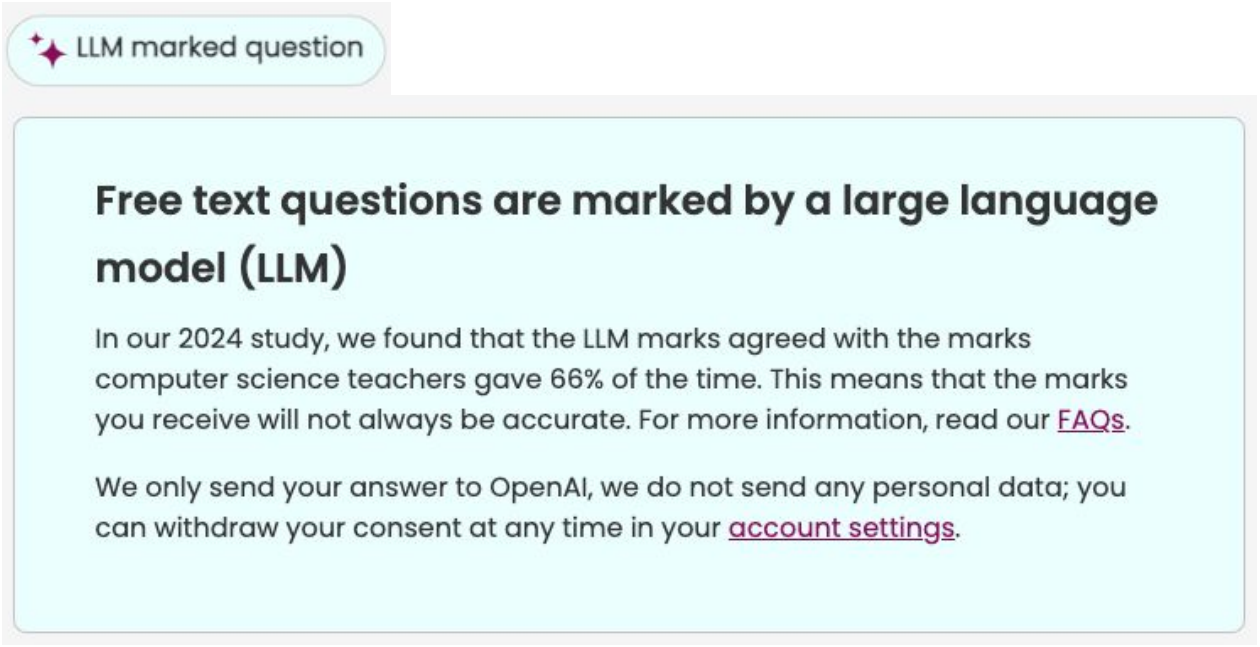
Self-marking questions

More than 1200 self-marking questions. Thirteen question types. Designed to help learners self-assess and reinforce learning.





LLM marking of free text questions



Questions are tagged as LLM marked

Banner with accuracy results from pilot study and further information

Example question: Environmental impacts

29 attempts remaining today

The rapid advancements in mobile phone technology have contributed to a culture of frequent upgrades, where consumers are encouraged to buy new phones regularly.

While this trend offers numerous benefits, it also comes with significant environmental consequences. Understanding both the positive and negative impacts of this upgrade culture on the environment is critical to fostering responsible consumer behaviour.

Give **one negative** and **one positive** environmental impact of the mobile phone upgrade culture.

[2 marks]

Disposing of mobile phones creates e-waste which can potentially release harmful chemicals into the environment. However, some devices use more energy efficient components which reduces the amount of power consumed.

Number of available marks displayed

Do you agree with the LLM's predicted marks?

1 in 3 times the predicted mark will be wrong. Find out more in our FAQs.

✨ **Prediction: 2 out of 2 marks**

Mark scheme	
Negative: Electronic waste (e-waste) increases as old phones are discarded.	✓
Negative: Mining for the materials needed to create devices causes habitat destruction.	
Negative: Manufacturing new devices consumes large amounts of fossil fuels.	
Negative: Manufacturing new devices releases carbon emissions and increases air pollution.	
Positive: There are recycling programmes for old phones.	
Positive: Newer phones may use more energy-efficient components compared to older phones.	✓
Positive: Some phones now have power-saving settings, e.g. displays that dim automatically when not being used.	

States marks are a prediction with purposeful, challenging language

Displays the full mark scheme (used by the LLM) and highlights the marks awarded

The prompt also includes a range of example responses with varying marks, which are not visible to users.

Before submitting another response, please say whether you agree with the predicted mark.

- [ ] Disagree
- [ ] Partly agree
- [ ] Agree

Send feedback

Users are encouraged to report on agreement of marks given

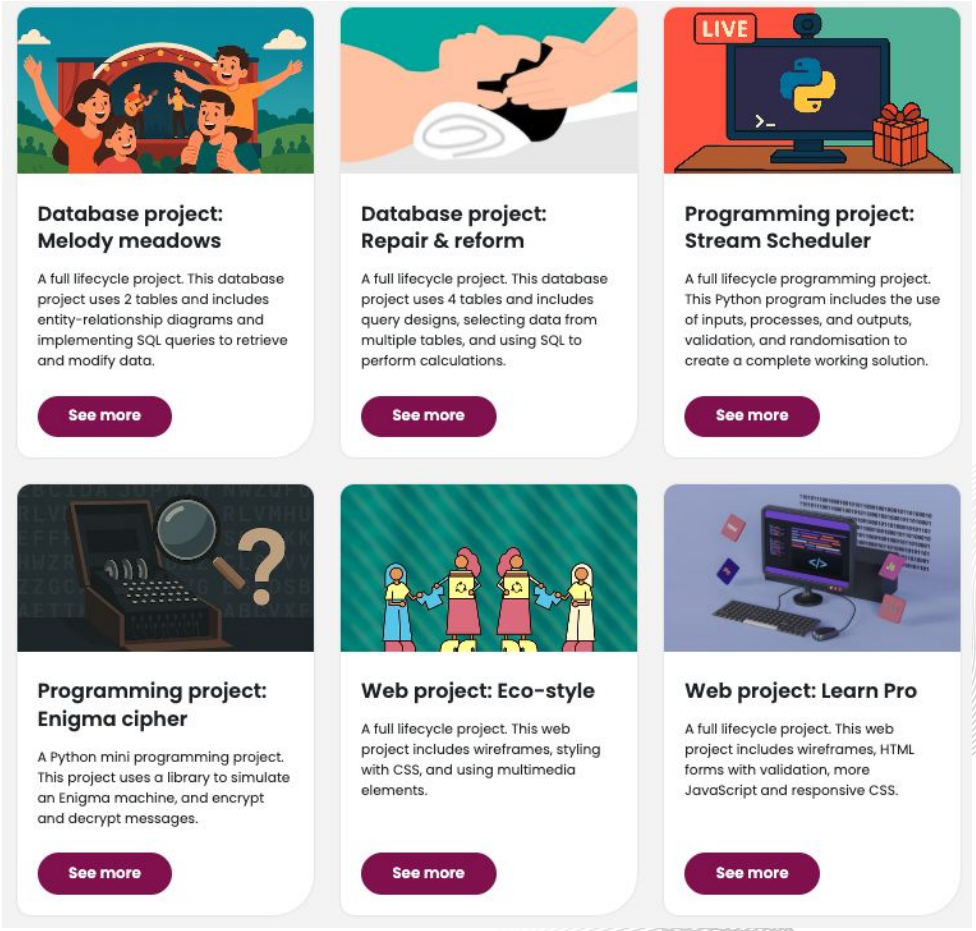
Projects

Projects on Ada CS

Projects are designed for students to work through at their own pace.

These projects help students build and develop practical computing skills.

Each project targets a specific set of skills, and is linked to relevant theory content to consolidate learning.

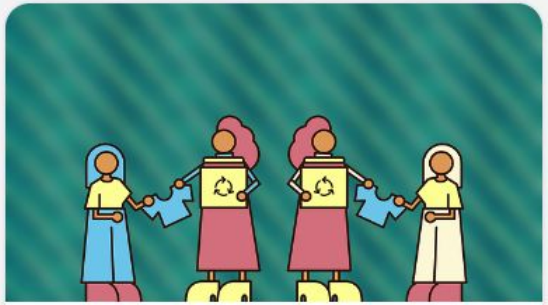


Web projects on Ada CS

We have produced two **web projects** that allows students to work through the whole system development life cycle and to support practical skills.

🔗 Eco-style

🔗 Learn Pro



Web project: Eco-style

A full lifecycle project. This web project includes wireframes, styling with CSS, and using multimedia elements.

See more

Web project: Learn Pro

A full lifecycle project. This web project includes wireframes, HTML forms with validation, more Javascript and responsive CSS.

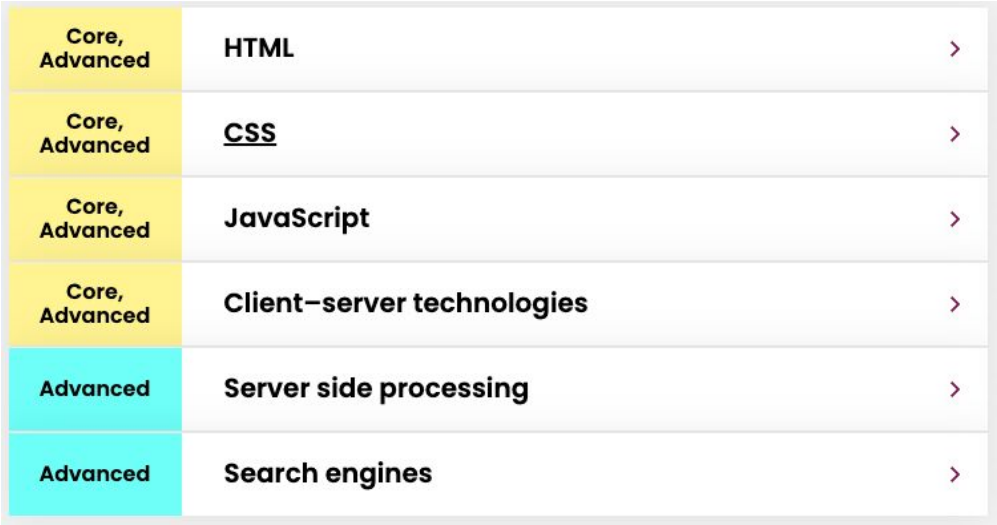
See more

Web technologies content on Ada CS

The topic pages on Ada CS have also been updated to include more information on web technologies (HTML, CSS and JavaScript).

Content for exam boards in England, Scotland and Wales.

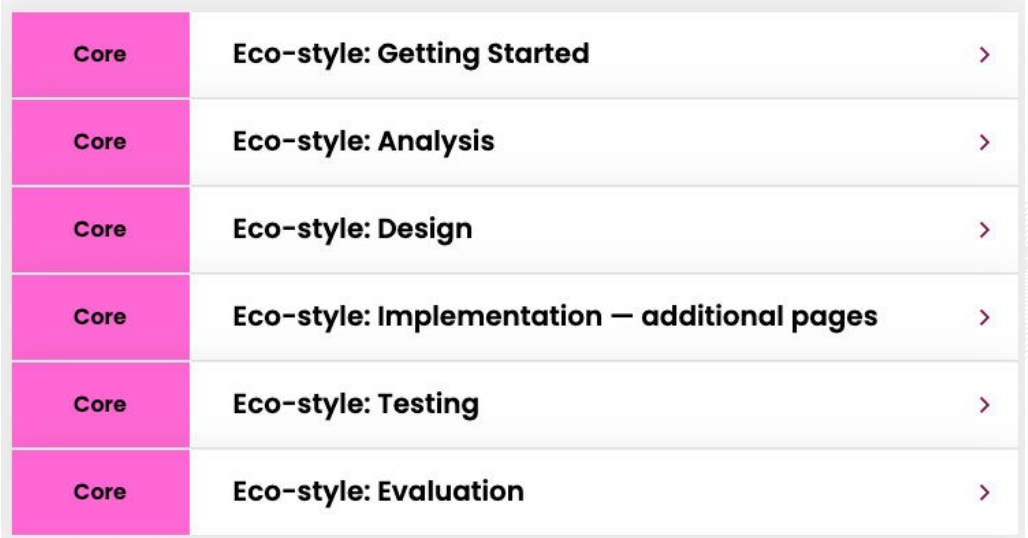
Set your viewing preferences so you see only what you want to see!



Core web project: Eco Style

This project is a full systems lifecycle project. Students will:

- Create a set of requirements based on the analysis of the Eco-style website.
- Design and create **2 web pages**.
- Style web pages using CSS.
- Create navigation between the webpages.
- Create and add multimedia to web pages.
- Test and evaluate their work



Examples of wireframes and step by step walk throughs

Example: Home page (index.html)

Figure 2 below is an example of a wireframe design for the existing home page of the Eco-style site.

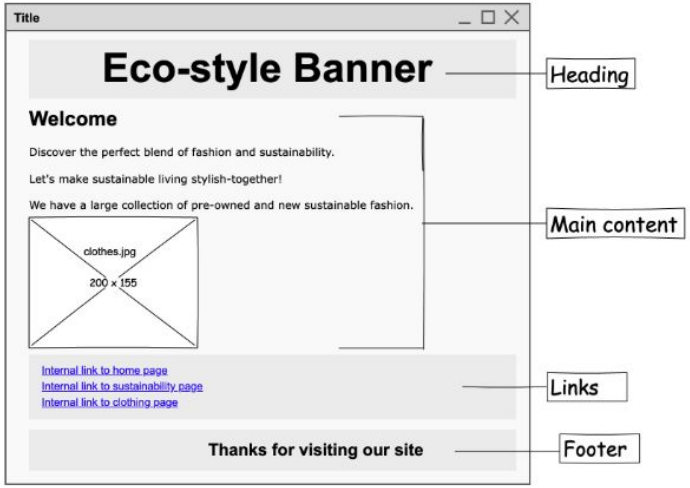


Figure 2: Wireframe for the home page

Adding div tags

Now you are going to add `<div>` tags to the **findUs.html** page.

The class attribute within a `<div>` tag is used to access a CSS class to the `<div>` element. This allows for consistent styling, organising content, and improving maintainability. How this works will be covered in the CSS sections later on.

Step 1

Open **findUs.html** in your chosen editor.

Step 2

The first `<div>` tag for the heading will go after the opening `<body>` tag.

```
<!DOCTYPE html>
<html>
  <head>
      <title> Find us </title>
  </head>
  <body>
  <div class ="heading">
  </div>
  </body>
</html>
```

Step 3

The remaining `<div>` tags will be placed within the body of the HTML page for the main content, links, and footer.

```
<!DOCTYPE html>
<html>
  <head>
      <title> Find us </title>
  </head>
  <body>
    <div class ="heading">
```

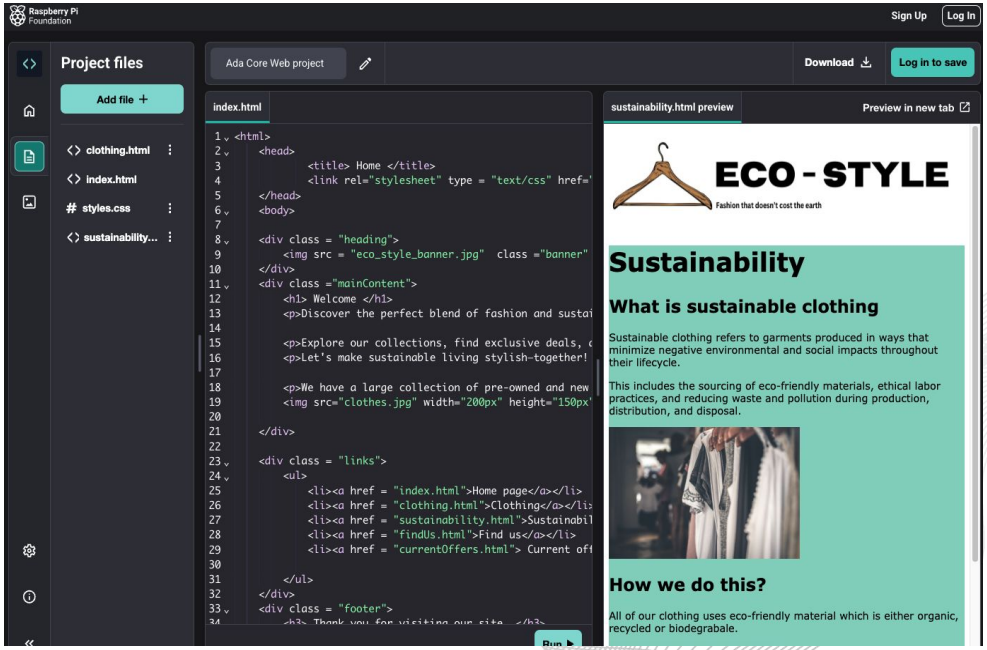
Code examples use online Raspberry Pi Code Editor

The examples on the site use the online **Raspberry Pi Code Editor** which supports Python and HTML/CSS.

Students can also download the starter files to use in the IDE of their choice.

We have preloaded the project with the starter website files for you here.

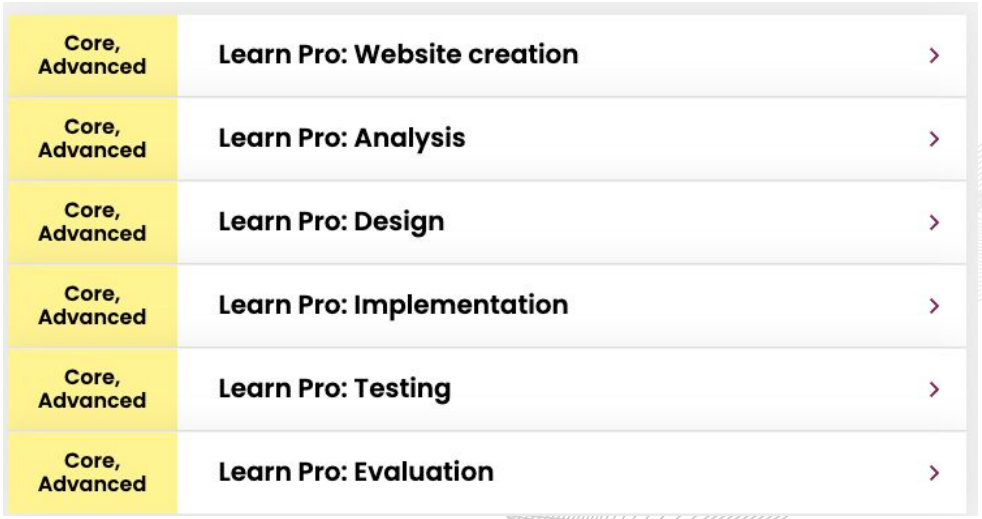
- Reduces barriers to getting started
- There are limitations of RPF code editor (as it is still in beta)



Intermediate web project: Learn Pro

This project is a full systems lifecycle project. The main differences between this and Eco Style is that students will:

- Prototype using low fidelity **wireframes** to develop an effective user-interface design.
- Design or complete **12 web pages** to build a multi-level website.
- Style web pages using **CSS** to control appearance and positioning using **floats** and **media queries**
- Use **HTML forms** and input **validation**
- Utilise Javascript in **onclick** and **onsubmit** events.



Learn Pro: Example functional requirements

Writing clear functional requirements is an area of weakness for most students.

The Intermediate project exemplifies writing clear, specific requirements.

Navigation

```
1. The website must have a clear top-level horizontal navigation bar with links to the
  1.1. Each of the 4 main pages will have links to its relevant sub-pages.
  1.2 Only subpages for the current main page should appear in the menu.
  1.3 Each active link should be highlighted in the menu.
  1.4 Each link should be highlighted when it is hovered over.
  1.5 The menu should change to a vertical navigation bar when viewed on smaller scree
  1.6 The company logo should appear at the top of every page and should take the user
  1.7 The footer should be consistent on each page and contain the contact information
```

Home page

```
2. The home page should be the "front window" for the company and have a pleasing layo
  2.1 There should be a welcome heading and a paragraph or two of welcome text describ
  2.2 There should be a promotional video on the right side of the screen.
  2.3 The video should autoloop and not show video controls.
  2.4 For each course category (Programming, Artificial Intelligence, Infrastructure),
  2.5 Each course category block should have a "Read more" link that takes the user to
  2.6 When the home page is viewed on a mobile device, the main heading, main text, vi
```

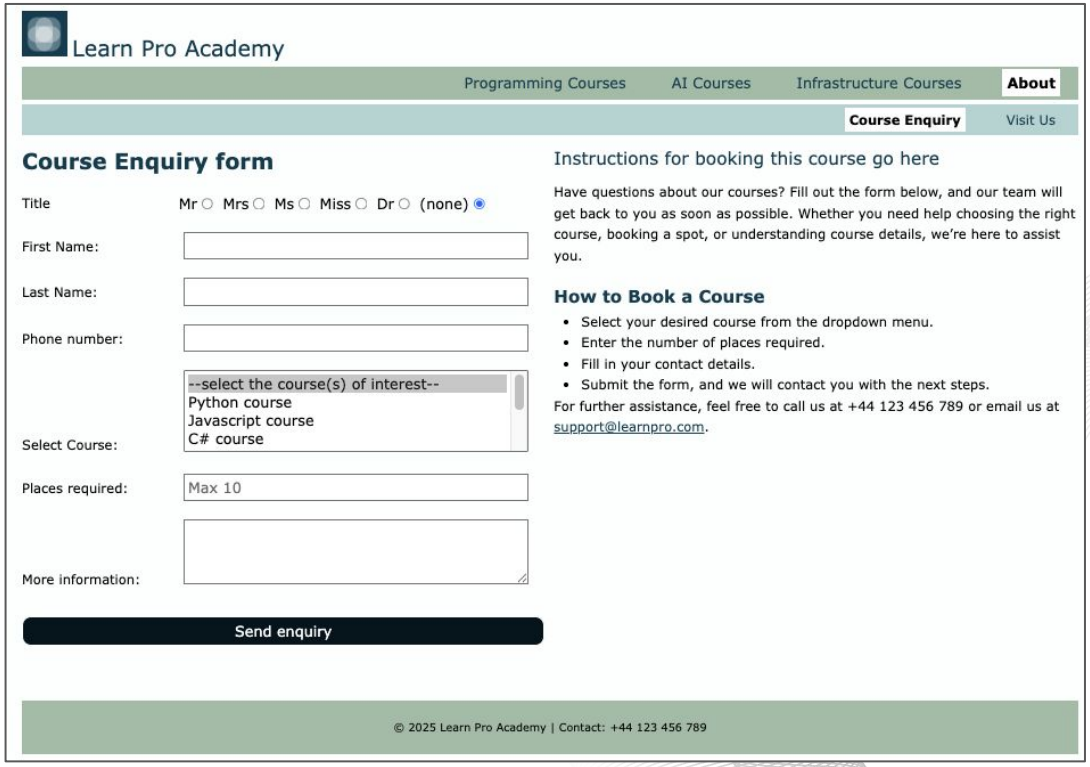
Course category pages

```
3. Each course category page should have information about the courses within each cat
  3.1  There should be a welcome heading and 1 or 2 paragraphs of welcome text describ
  3.2 There should be an image or video on the right side of the screen.
  3.3 Each course in this category should show a smaller image and text with a link wi
  3.4 Each course block should have a "Read more" link that takes the user to the appr
  3.5 When the course category page is viewed on a mobile device, the main heading, ma
```

Learn Pro: HTML Forms with validation

The intermediate project includes building an HTML form with different input types.

The form also includes JavaScript in the onsubmit event.



Database projects on Ada CS

We have produced two **database projects** that allows students to work through the whole system development life cycle and to support practical skills.

🔗 Melody meadows

🔗 Repair & reform

Database project: Melody meadows

A full lifecycle project. This database project uses 2 tables and includes entity-relationship diagrams and implementing SQL queries to retrieve and modify data.

See more

Database project: Repair & reform

A full lifecycle project. This database project uses 4 tables and includes query designs, selecting data from multiple tables, and using SQL to perform calculations.

See more

Database and SQL content on Ada CS

Core	Music festival: Analysis ›
Core	Music festival: Design ›
Core	Music festival: Database creation ›
Core	Music festival: Query implementation ›
Core	Music festival: Testing
Core	Music festival: Evaluation

Core, Advanced	SQL fundamentals ›
Advanced	SQL: CREATE and DROP statements ›
Core, Advanced	SQL: SELECT statements ›
Core	SQL: INSERT, UPDATE and DELETE statements ›

Updates to content made based on projects

Programming projects on Ada CS

We have recently launched our first full systems lifecycle **Python programming project** to support practical skills.

🔗 Stream Scheduler

Core	Stream Scheduler: Getting Started ›
Core	Stream Scheduler: Analysis ›
Core	Stream Scheduler: Design ›
Core	Stream Scheduler: Implementation ›
Core	Stream Scheduler: Testing ›
Core	Stream Scheduler: Evaluation ›

Programming project: Stream Scheduler

A full lifecycle programming project. This Python program includes the use of inputs, processes, and outputs, validation, and randomisation to create a complete working solution.

See more

Integrated project

This project utilises **server-side scripting** (PHP) and connecting to a **database** in MySQL.

The scenario is a tech course company - Course Pal.

🔗 Course Pal

- Database with **5 tables**
- **17 PHP driven pages** e.g.
  - Login & register routines
  - Course search
  - Course booking
- Multi-table SQL queries
- Plus **full systems lifecycle write-up**
- Guide to setting up a web server using a PHP / MySQL stack

Web project: Course Pal

A full systems lifecycle project which integrates PHP with a 5-table relational database and includes wireframes, UML diagrams, session variables, and responsive CSS.

See more

🔗 Course Pal

Full PHP/MySQL project walkthrough

A Level	Course Pal: Getting started ›
A Level	Course Pal: Analysis ›
A Level	Course Pal: Design ›
A Level	Course Pal: Implementation - Creating dynamic pages with a database ›
A Level	Course Pal: Implementation - Viewing data ›
A Level	Course Pal: Implementation - Editing data ›
A Level	Course Pal: Implementation - Data analysis ›
A Level	Course Pal: Testing ›
A Level	Course Pal: Evaluation ›

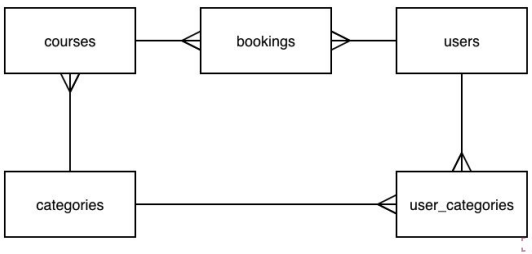


Figure 4: An entity–relationship diagram for the Course Pal project showing five tables and their relationships

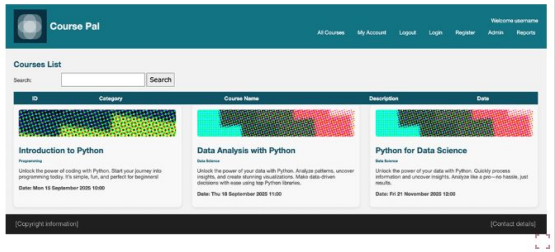
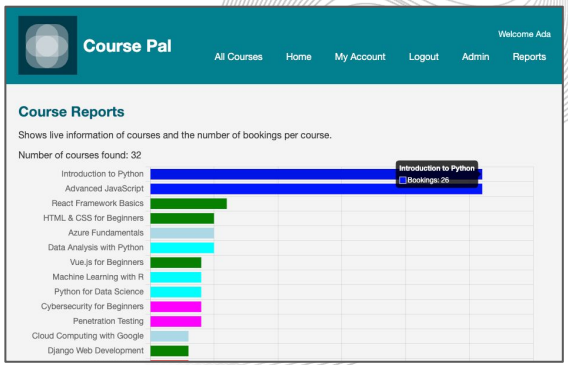
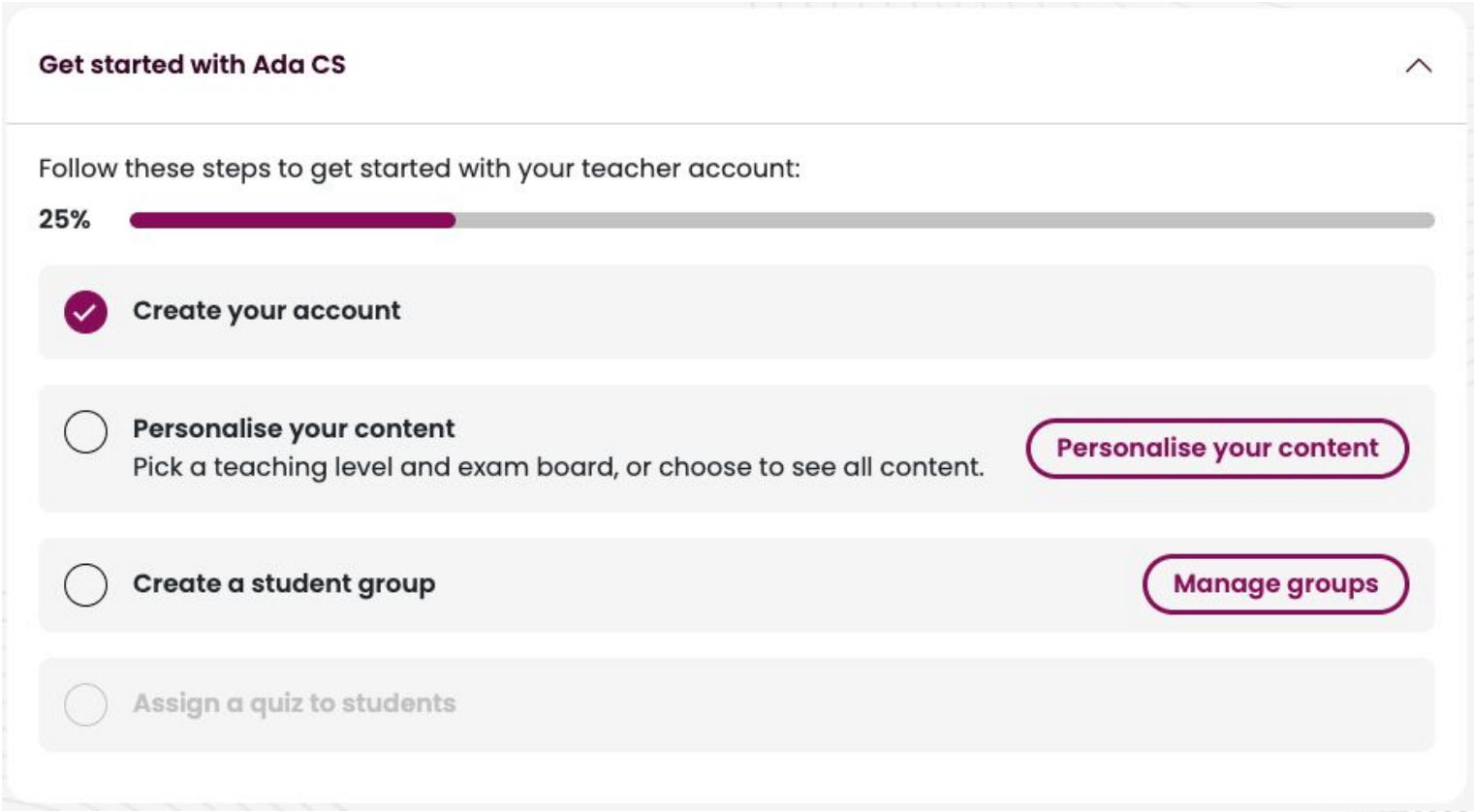


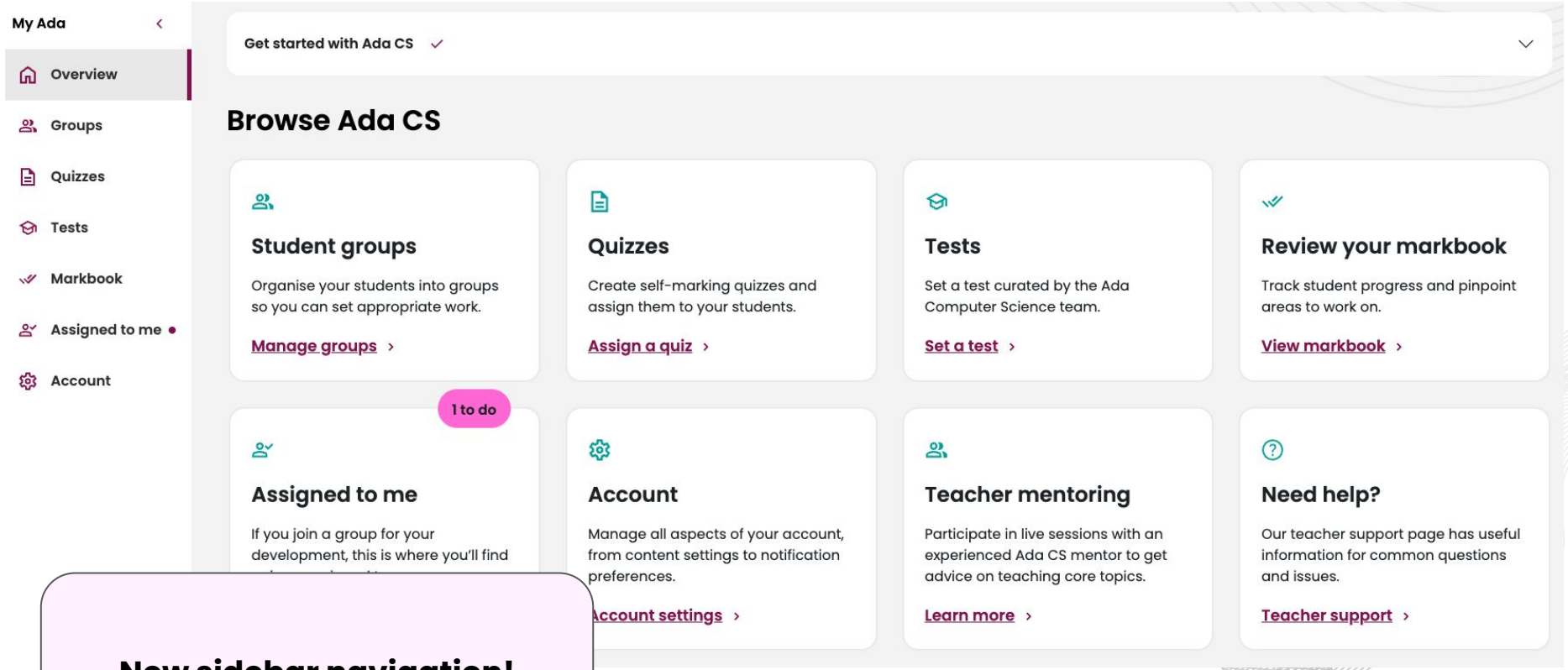
Figure 14: The **courses.php** page showing the three courses that contain the word "Python"



Class management tools

Teacher onboarding [New!]




My Ada
Overview
Groups
Quizzes
Tests
Markbook
Assigned to me
Account
Get started with Ada CS
Browse Ada CS
Student groups
Organise your students into groups so you can set appropriate work.
Manage groups
Quizzes
Create self-marking quizzes and assign them to your students.
Assign a quiz
Tests
Set a test curated by the Ada Computer Science team.
Set a test
Review your markbook
Track student progress and pinpoint areas to work on.
View markbook
1 to do
Assigned to me
If you join a group for your development, this is where you'll find
Account
Manage all aspects of your account, from content settings to notification preferences.
Account settings
Teacher mentoring
Participate in live sessions with an experienced Ada CS mentor to get advice on teaching core topics.
Learn more
Need help?
Our teacher support page has useful information for common questions and issues.
Teacher support
New sidebar navigation!

Manage groups

You can add other teachers to help you manage a group. You cannot directly add students, but you can invite them to join.

Create a new group

Groups: Alphabetical

Active Archived

- Demo Group 259
- Demo Group 867
- Demo Group 1330
- Demo Group 1453
- Demo Group 1549
- Demo Group 1550

Group details

Rename group

Demo Group 259 Update

Group managers

Edit group managers

Group Manager 1 (group owner) (you)
Group Manager 2

Group members

Invite users Email users

Allow students to remove themselves from this group

This group has 3 members.

Test Student Reset password Remove

Manage assignments

Our self-marking quizzes help you to check your students' knowledge. You can create your own or choose a pre-made quiz.

Set a quiz as an assignment to see student performance in your markbook. You can share a link to a quiz without assigning it, but you won't see how students perform.

Your quizzes

You have 53 created quizzes.

Create a quiz | View pre-made quizzes

Display in: Table View

Filter quizzes by name:

Filter by Creator: Me

Groups	Quiz name	Stages and Difficulties	Creator	Last viewed	Manage	Share	Delete
0 groups	Searching algorithms	GCSE (P1, P2, C1, C2) A Level (P1, P2, C1, C2) Adv Higher (P1, P2, C1, C2)	Me	06/05/2025	Assign		

Markbook [New UI!]

Download CSV

Due: 19/01/2025 | 29 of 81 attempted all questions | 26 of 81 got full marks

Group overview | Detailed marks

Group assignment overview

See who attempted the assignment and which questions they struggled with.

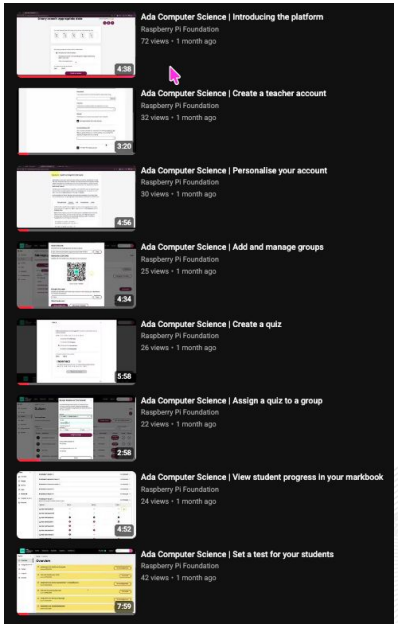
Show mark as percentages

Key: Correct | Partially correct | Incorrect | — Not attempted

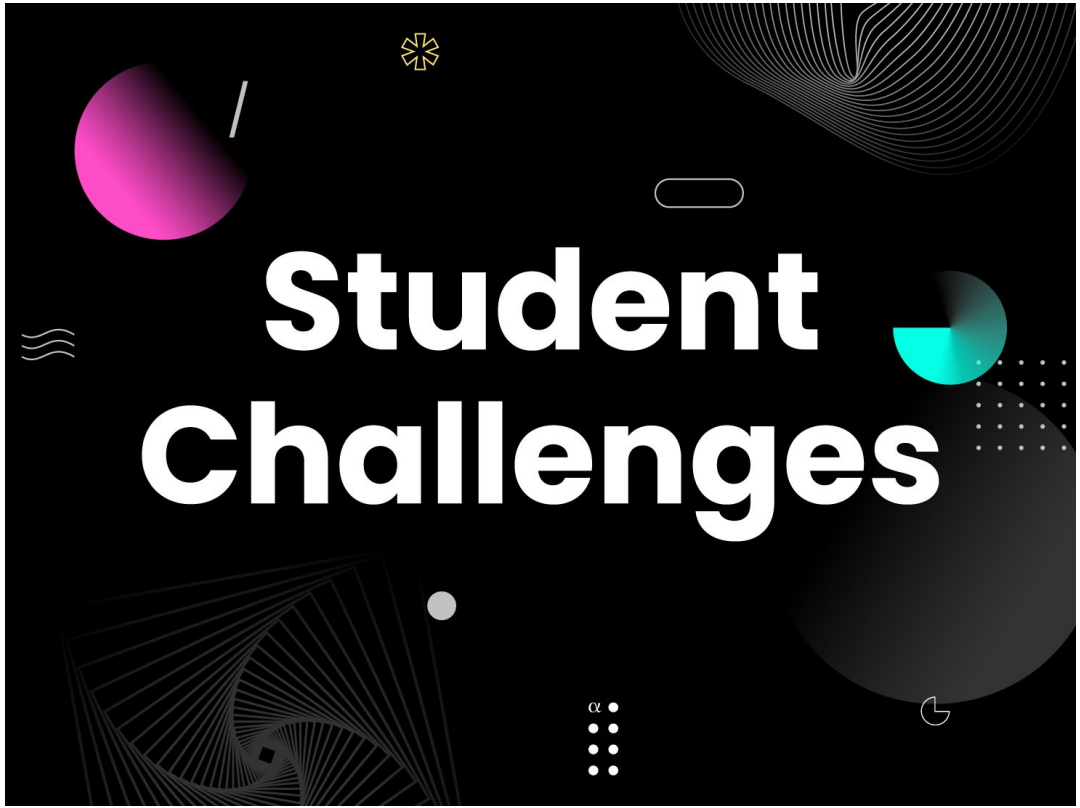
Name ▲	Correct ▲	1 ▲	2 ▲	3 ▲	4 ▲	5 ▲	6 ▲	7 ▲	8 ▲	9 ▲	10
Test Student 29	10/10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Test Student 3	7/10	✓	✓	✓	✓	✓	✓	✓	—	—	-
Test Student 30	8/10	✓	✓	✗	✓	—	✓	✓	✓	✓	✓
Test Student 31	10/10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Getting started with Ada Computer Science (Youtube Playlist)

rpf.io/adacs-get-started



Other support



Autumn term
Cryptography/ cyber based challenge

Spring term
Coding challenge

Summer term
Revision challenge

🔗 Student challenges

Suggested teaching order

Core Y1 | Core Y2 | Advanced Y1 | Advanced Y2

The content for formal computer science qualifications varies significantly, and different exam boards require students to learn different material.

On this page, we suggest an order in which you may like to address topics, especially if you are not delivering the content for a formal qualification. We hope this suggested order will provide an effective path through the material and be useful to educators new to the subject.

If you teach a specific qualification, your exam board may publish a suggested teaching order specific to that qualification. You can view the specification pages for England, Scotland and Wales here.

This page covers the **core concepts** of computer science typically studied by students aged 14-16, as well as the **advanced concepts** typically studied by students aged 16-19.

Term 1	
Theory	**Programming**
Hardware	Computational thinking
Software	Data types, variables, constants, inputs

Teacher mentoring

This programme is designed to support **newly qualified or non-specialist** computer science teachers. It will cover the topics that are found on most GCSE and A level (ages 14-19) computer science specifications. However, the content should be relevant to teachers of other qualifications and at other levels.

New! how to video clips: rpf.io/adacs-get-started

More than just Ada CS!

STEMSMART
An Initiative from the University of Cambridge

- The Ada CS team delivers the computer science strand of the 🔗 **STEM SMART** programme alongside our colleagues at the University of Cambridge.
    - Aimed at supporting able but disadvantaged students apply to top universities.
    - Students all use Ada CS to support their studies.
- We deliver the 🔗 **Bebras Challenge** and the **Coding Challenge** in the UK.
    - Bebras is the world's largest computational thinking competition; this year over 460,000 students took part in the UK.
    - We are adding Bebras style questions on Ada and links to Ada learning material on the Bebras site.


ada
computer
science

Thank you!


UNIVERSITY OF
CAMBRIDGE


Raspberry Pi
Foundation