What is Computer Science? What does it try to understand about the world? And why is it important to become a computer science teacher in today’s world? Through the prompts in this journal you can direct and document your learning as a computer science teacher.

The lesson reflections at the end of this journal offer you possibilities to recall, revisit and interpret your learning. You can fill the pages as you progress through the videos in order, or jump around.
One area of computer science I’m curious about

Subject area I’m most excited about integrating with CS

My goal after this day as a computer science teacher is to

One magical childhood experience
What resources are available for teaching computer science? (E.g. books, laptops, tablets, robots, crafts materials..)

When will you be teaching computer science? Who else will be teaching computer science in your school?

How could you engage parents, industry and other stakeholders?

Identify learning communities in computer science for on-going support

How will I be successful in teaching computer science?

What is my teaching style?

What will be a challenge for me teaching computer science?

What will I need to know before implementing?
Session 1

MY LOVE LETTER TO COMPUTERS

Write a letter for the principal, parents, local media or another important stakeholder on the importance of computer science education. Highlight what computer science is, why it’s important, how and where you will begin integrating it and what kind of support you’d need.

Fill this page after watching the first video on Computer Science.
Dear ______,


Choose one of the concepts or practices of computational thinking and explore more in-depth.

What does it mean?

How it applies to your work?

How else you could you use it?
Explain the symbol used in the title

Make up an action/dance/poem/rap/song to represent the idea

Describe how the idea can influence your other subjects

Fill this page after watching the Computational Thinking video.
Session 3

Get to Know a Curriculum

Get to know one coding program or curriculum better and reflect on how you might use it in your classroom.

Curriculum Title

Provider

What prior knowledge will a teacher need?

What prior knowledge will a student need?

What student outcomes can be met with this course?

Fill after watching the third video on programming.
How long will this course take to implement?

What about this course would be a success in your classroom?

What about this course would be a challenge in your classroom?

What equipment or resources would you need to implement this course?

What examples of sequence, selection and iteration did you find?

Who else is using this curriculum?

See helloruby.com/loveletters for a list of coding resources
Find a Computer Science quote that is interesting to you and copy it here

“COMPUTERS ARE THE BICYCLE FOR THE MIND”
- STEVE JOBS

How could you make computer science visible in your classroom?

What kind of books could your classroom library include? What about posters or bulletin boards?

Determine at least two ways to include your students work around the classroom.
You can use Hello Ruby characters by going to helloruby.com/scratch or use the existing Scratch characters of the platform.
Session 6

Mental Models

Draw what you imagine is inside a computer. Then elaborate around this initial mental model.

Does your drawing present concepts or surface detail?

What kind of ways do you learn?

Fill this page after watching the sixth video on Hardware.
How could you use different mediums to explore ideas around computer science? Pick from the suggestions or choose your own and brainstorm. Use Memo space on next page.

<table>
<thead>
<tr>
<th><strong>I/O systems</strong></th>
<th><strong>Hardware</strong></th>
<th><strong>Algorithm</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Paint</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Draw</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stickers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clay</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Play</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dough</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wire</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Photography</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Puppets, Masks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Session 8

ASSESSMENT

How would you go about assessing students. Choose one activity you’ve completed and brainstorm different approaches.

**Reflection.** What guiding questions could you offer for students to reflect their work? How about peer feedback?

**Worksheets and quizzes.** How could you apply these in the context of computer science?
- Ask students to predict what a piece of code does. What happens next?
- Ask students to reverse engineer a project, and explain what kind of code might be used to make it.
- Ask students to remix or debug a project.

**Rubrics and checklists.** Can you make a rubric to support your teaching?

**Portfolio.** What kind of portfolio project could the students build?

After watching the eight video on Internet.
Excitement.
Which aspect of computer science are you particularly curious or excited about and why?

Worries.
What worries you about computer science?

Needs.
What else do you need to find out about this topic?

Stance, Steps, or Suggestions.
What should be your next steps? What suggestions do you have at this point?

after watching the ninth video on machine learning and AI.
Read through the Myths in Computing Education. Which one do you agree with? Which one do you disagree with? Reach out to one or more peers and discuss together, online or offline. Then read through the original article by Mark Guzdial.

<table>
<thead>
<tr>
<th>Myth</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The lack of women in Computer Science is just like all the other STEM fields.</td>
<td>To get more women in CS, we need more female CS faculty.</td>
</tr>
<tr>
<td>Good teachers personalize education for students’ learning styles.</td>
<td>A good CS teacher should model good software development practices because their job is to produce excellent software engineers.</td>
</tr>
<tr>
<td>To get more women in CS, we need more female CS faculty.</td>
<td>Student evaluations are the best way to evaluate teaching.</td>
</tr>
<tr>
<td>A good CS teacher should model good software development practices because their job is to produce excellent software engineers.</td>
<td>Some people are just naturally better programmers than others.</td>
</tr>
</tbody>
</table>

“I used to think          but now I think                         ”

How can you introduce a more diverse idea of computer science in your classroom?

List three businesses, individuals or organisations you could invite for a classroom visit.
### WHAT Did I Learn?

**Do a final assessment of your skills. Choose the right emoji:**

**How did it go?**

I can explain what an algorithm is in the context of computer science. | I Can Do It! | I Did it but didn’t fully get it | I Didn’t get it at all |
---|---|---|---|
I can give examples of algorithms met in everyday life. | | | |
I can explain that computers are controlled by sequences of precise instructions known as programs. | | | |
I can explain how computers use input, process and output to carry out useful tasks. | | | |
I can program a simple computer game using Scratch. | | | |
I can explain basic programming concepts to children (e.g., algorithms, loops, conditionals, functions). | | | |
I can plan out the logic for a computer program even if I don’t know the specific programming language. | | | |
I know where to find the resources to help students learn to code. | | | |
I can find applications for coding that are relevant for students. | | | |
I can help students debug their code. | | | |
I can explain what a computer is and give examples of devices that include computers. | | | |
I can describe the key characteristics of basic computer architecture (e.g, CPU, memory, hard disk, mouse, display etc). | | | |
I can explain in simple terms what a computer network is. | | | |
I can discuss social and ethical issues raised by the role of computers in the world. | | | |
I can suggest career paths for those studying Computing. | | | |
I have ideas on how to integrate computer science into my classroom. | | | |
I can plan, create and assess creative computing curriculum. | | | |

**New goal**

---

**I Can Do It!**

I can explain what an algorithm is in the context of computer science.

I can give examples of algorithms met in everyday life.

I can explain that computers are controlled by sequences of precise instructions known as programs.

I can explain how computers use input, process and output to carry out useful tasks.

I can program a simple computer game using Scratch.

I can explain basic programming concepts to children (e.g., algorithms, loops, conditionals, functions).

I can plan out the logic for a computer program even if I don’t know the specific programming language.

I know where to find the resources to help students learn to code.

I can find applications for coding that are relevant for students.

I can help students debug their code.

I can explain what a computer is and give examples of devices that include computers.

I can describe the key characteristics of basic computer architecture (e.g, CPU, memory, hard disk, mouse, display etc).

I can explain in simple terms what a computer network is.

I can discuss social and ethical issues raised by the role of computers in the world.

I can suggest career paths for those studying Computing.

I have ideas on how to integrate computer science into my classroom.

I can plan, create and assess creative computing curriculum.

---

**I Did it but didn’t fully get it**

---

**I Didn’t get it at all**

---

**I Can’t Do It!**

---
Choose one practice and one concept you want to teach.

- Decomposition
- Data
- Pattern recognition
- Algorithms
- Hardware
- Something else:

- Persistence
- Debugging
- Tinkering
- Collaboration
- Abstractions
- Automation
- Something else:

**1. SENSE AND EXPERIENCE**
Choose a learning activity that immerses students in the new concept.

**2. RECORD AND RETAIN**
Reflect on the experience, through discussing, writing and drawing, for example.

**3. MANIPULATE AND EXPERIMENT**
Create an experiment for the learners to run - this could be planned in advance or could grow out of the students’ engagement with the earlier activity.

**4. FRAME ACADEMICALLY OR TECHNICALLY**
Now, students can deepen their knowledge of the subject through listening to or reading more technical academic texts. What resources will you use?
SYNTHESIS AND PRODUCE
All the previous learning experiences and knowledge gained are combined into a product with a clear recipient.

PREP TIME!
What materials, requirements or other preparation your lesson plan requires?

ASSESSMENT
How do you check students understanding?

TEACHER CREATED
STUDENT CREATED
TEACHER ASSESSED
STUDENT ASSESSED

FEEDBACK
Ask from three other participants feedback on your project idea.

What is something that works well or you really like about the project?
What is something that is confusing or could be done differently?
What is something that doesn’t work or could be improved?
COMPUTATIONAL THINKING: LESSON REFLECTION

Date

What went well?

What didn’t go well?

How did the students respond?

Any special moments with students?

How could I improve this lesson next time?

Was the objective met? Why or why not?

Next steps
Date

What went well?

What didn’t go well?

How did the students respond?

Any special moments with students?

Was the objective met? Why or why not?

How could I improve this lesson next time?

Next steps
Date

What went well?

What didn’t go well?

How did the students respond?

Any special moments with students?

Was the objective met? Why or why not?

How could I improve this lesson next time?

Next steps
AI and Machine Learning:
Lesson Reflection

Date

What went well?

What didn’t go well?

How did the students respond?

Any special moments with students?

How could I improve this lesson next time?

Was the objective met?
Why or why not?

Next steps
Diversity and Equity: Lesson Reflection

Date

What went well?

What didn’t go well?

How did the students respond?

Any special moments with students?

How could I improve this lesson next time?

Was the objective met? Why or why not?

Next steps