Lesson 1: Computational Thinking Basics

Scott Morgan

Bridgend College
BTEC Computing: Computational Thinking (Unit 18)

Web: scott3142.com
E-mail: MorganSN@cardiff.ac.uk

Computers Are Stupid!

- Computers need to be *programmed*.

Computers Are Stupid!

- Computers need to be *programmed*.
- They cannot (yet) think for themselves.

Computers Are Stupid!

- Computers need to be *programmed*.
- They cannot (yet) think for themselves.
- They speak a special language - *binary*.

Computers Are Stupid!

- Computers need to be *programmed*.
- They cannot (yet) think for themselves.
- They speak a special language - *binary*.
- They are highly *logical*.

Recipes and Algorithms

- Programs are like *recipes*.

RECIPE: Tomato vegetable pasta

Serves: 5

Cost per portion: 32p

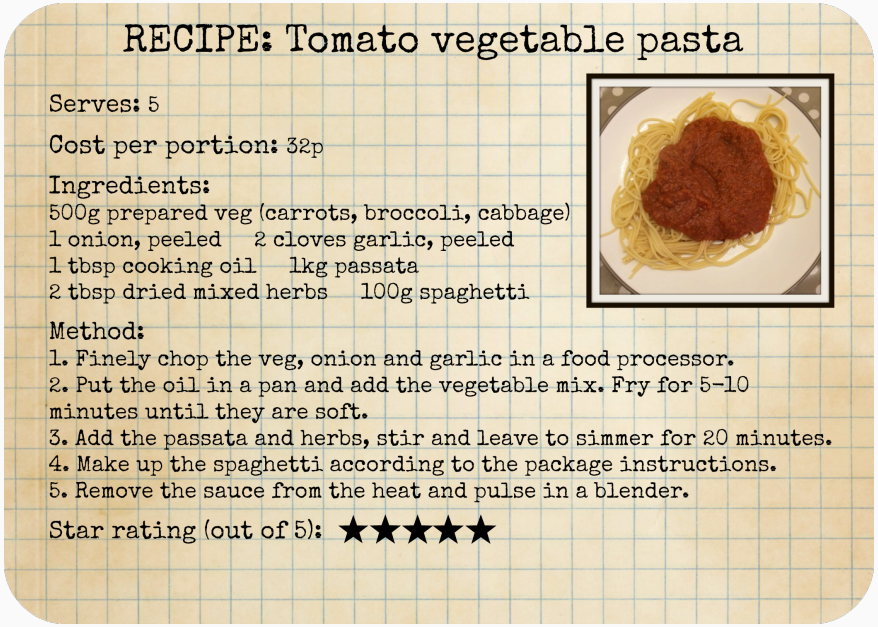
Ingredients:
500g prepared veg (carrots, broccoli, cabbage)
1 onion, peeled 2 cloves garlic, peeled
1 tbsp cooking oil 1kg passata
2 tbsp dried mixed herbs 100g spaghetti

Method:
1. Finely chop the veg, onion and garlic in a food processor.
2. Put the oil in a pan and add the vegetable mix. Fry for 5–10 minutes until they are soft.
3. Add the passata and herbs, stir and leave to simmer for 20 minutes.
4. Make up the spaghetti according to the package instructions.
5. Remove the sauce from the heat and pulse in a blender.

Star rating (out of 5): ★★★★★

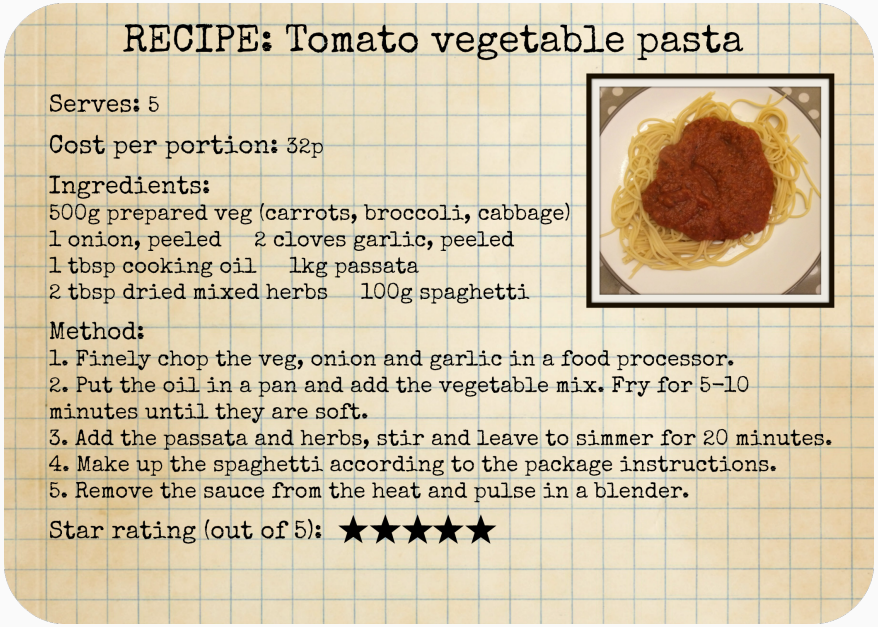
Recipes and Algorithms

- Programs are like *recipes*.
- Each instruction must lead to the next one.



Recipes and Algorithms

- Programs are like *recipes*.
- Each instruction must lead to the next one.
- Nothing can be *assumed*!



Recipes and Algorithms

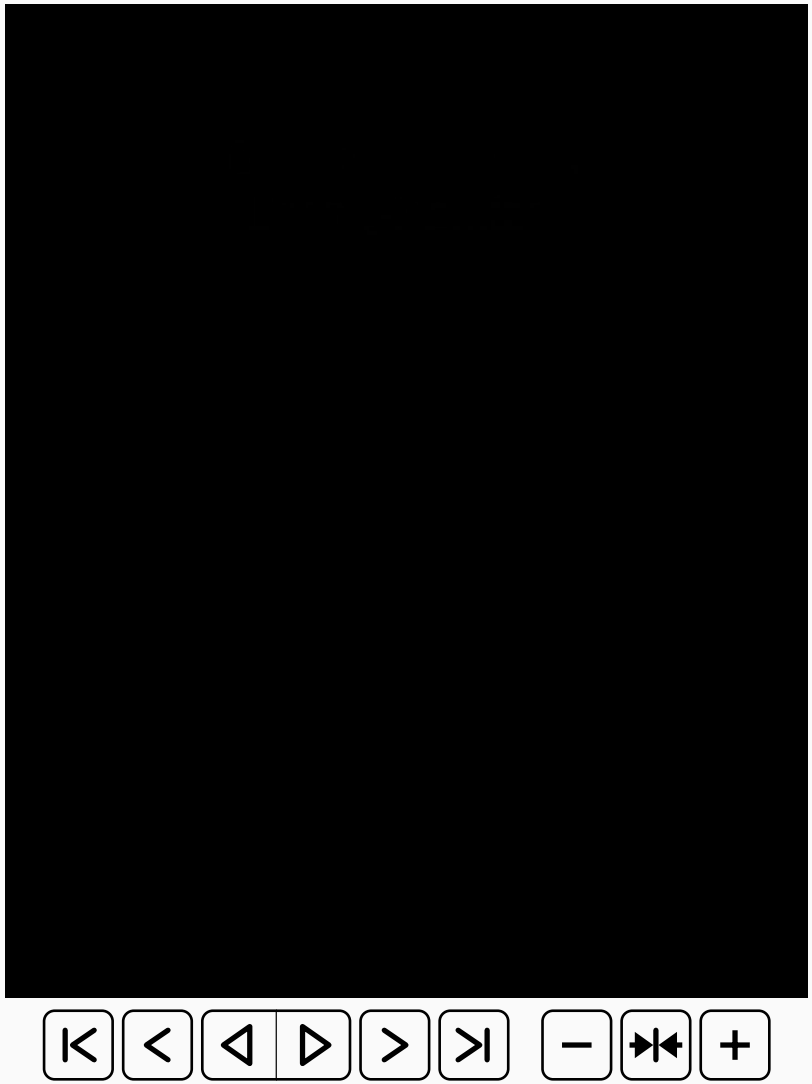
- Programs are like *recipes*.
- Each instruction must lead to the next one.
- Nothing can be *assumed*!

What does this do?

```
1 N = 10
2
3 for j in range(1,N):
4     if j == 1:
5         print(j-1)
6     else
7         print(j)
8
```

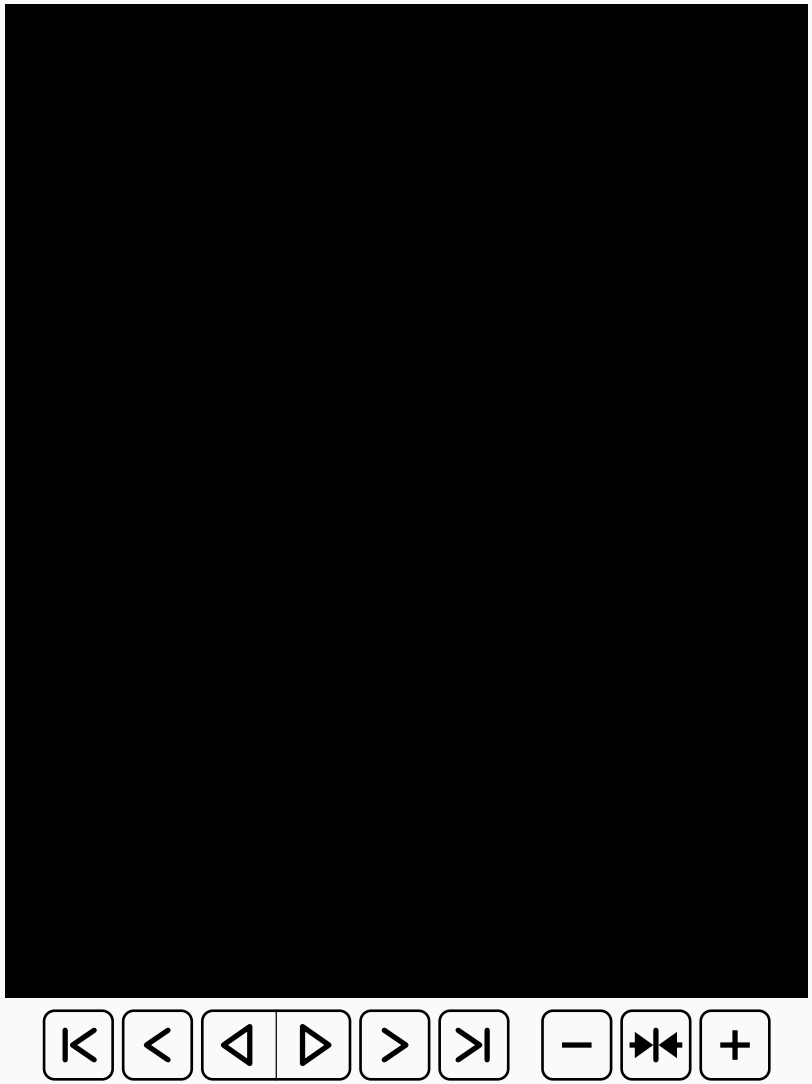
Computers Need Rules

- Computers take an input, follow some rules, and give an output.



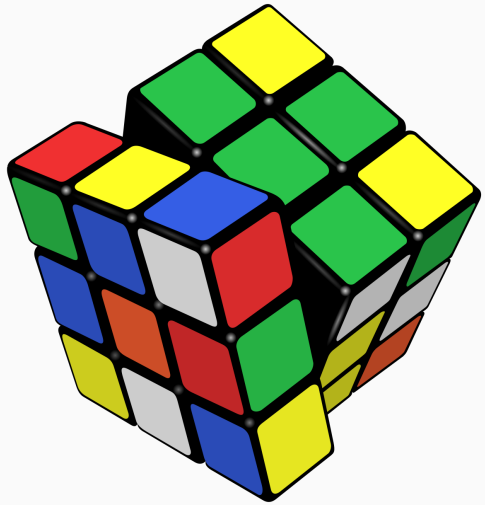
Computers Need Rules

- Computers take an input, follow some rules, and give an output.
- Based on minimal input, computers can be *trained* to excel at certain tasks.



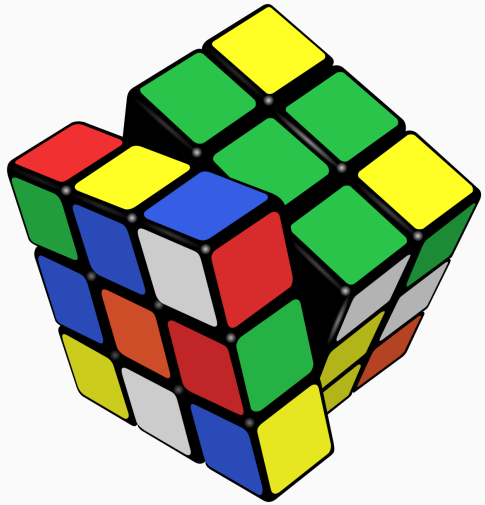
Algorithmic Thinking

- Break large problem into small tasks.



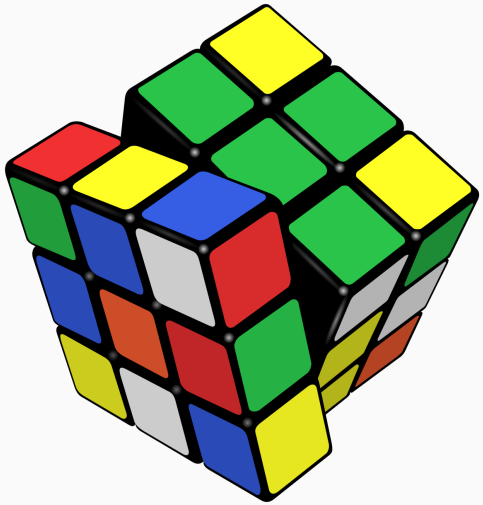
Algorithmic Thinking

- Break large problem into small tasks.
- Assemble small tasks to solve large problem.



Algorithmic Thinking

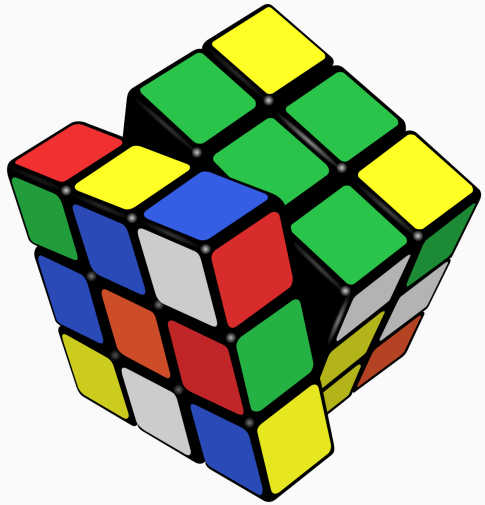
- Break large problem into small tasks.
- Assemble small tasks to solve large problem.



1. Solve top white face.

Algorithmic Thinking

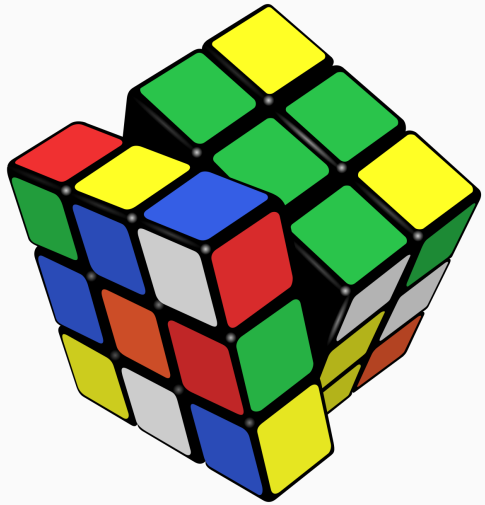
- Break large problem into small tasks.
- Assemble small tasks to solve large problem.



1. Solve top white face.
2. Position corner pieces.

Algorithmic Thinking

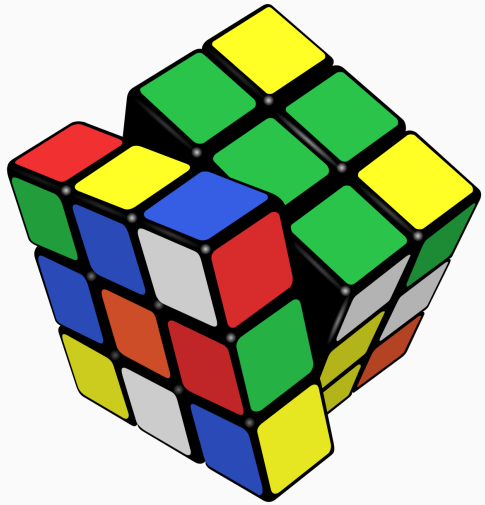
- Break large problem into small tasks.
- Assemble small tasks to solve large problem.



1. Solve top white face.
2. Position corner pieces.
3. Solve side faces.

Algorithmic Thinking

- Break large problem into small tasks.
- Assemble small tasks to solve large problem.



1. Solve top white face.
2. Position corner pieces.
3. Solve side faces.
4. Solve underside.

Bugs are Everywhere!

- *Debugging* is one of the most important parts of programming.

Bugs are Everywhere!

- *Debugging* is one of the most important parts of programming.
- To get good at debugging, you have to *play*! Make mistakes, fix them, make more mistakes.

Bugs are Everywhere!

- *Debugging* is one of the most important parts of programming.
- To get good at debugging, you have to *play*! Make mistakes, fix them, make more mistakes.
- **Making mistakes is how you learn here! No one (EVER) gets it right the first time!**

Read - Search - Ask

Read - Search - Ask

Read

- Read the documentation.

Search

Ask

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?

Search

Ask

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?
- Read the API (Application Programming Interface).

Search

Ask

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?
- Read the API (Application Programming Interface).

Search

- Google is your friend!

Ask

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?
- Read the API (Application Programming Interface).

Search

- Google is your friend!
- More often than not, someone has already had your problem!

Ask

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?
- Read the API (Application Programming Interface).

Search

- Google is your friend!
- More often than not, someone has already had your problem!
- They've usually asked it on StackExchange.

Ask

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?
- Read the API (Application Programming Interface).

Search

- Google is your friend!
- More often than not, someone has already had your problem!
- They've usually asked it on StackExchange.

Ask

- **This is so important!**

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?
- Read the API (Application Programming Interface).

Search

- Google is your friend!
- More often than not, someone has already had your problem!
- They've usually asked it on StackExchange.

Ask

- **This is so important!**
- Collaboration and sharing ideas is how this field advances.

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?
- Read the API (Application Programming Interface).

Search

- Google is your friend!
- More often than not, someone has already had your problem!
- They've usually asked it on StackExchange.

Ask

- **This is so important!**
- Collaboration and sharing ideas is how this field advances.
- Ask your friends, your teachers, the Internet.

Read - Search - Ask

Read

- Read the documentation.
- Does your program have a README file?
- Read the API (Application Programming Interface).

Search

- Google is your friend!
- More often than not, someone has already had your problem!
- They've usually asked it on StackExchange.

Ask

- **This is so important!**
- Collaboration and sharing ideas is how this field advances.
- Ask your friends, your teachers, the Internet.
- Use the *gitter* chat for this!

Task

Experimenting with Python (15 mins)

Follow the instructions for Computaitonal Thinking - Lesson 1 at scott3142.com/btec

---

Experiment with the notebook. Change the variable names. Change the numbers. Play around. Make mistakes and fix them. Remember **Read-Search-Ask***!*

Formulaic Thinking

You will come across lots of formulae in this course!

Formulaic Thinking

You will come across lots of formulae in this course!

Don't freak out if you see them!

Formulaic Thinking

You will come across lots of formulae in this course!

Don't freak out if you see them!

Approach them like a computer would!

Formulaic Thinking

You will come across lots of formulae in this course!

Don't freak out if you see them!

Approach them like a computer would!

- *Break down into small pieces that are easy to manage.*
- *Think logically!*

Formulaic Thinking

- What is the point of formulae?

$$y = mx + c$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Formulaic Thinking

- What is the point of formulae?
- Where are they used in computing?

$$y = mx + c$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$E = mc^2$$

$$e^{i\pi} + 1 = 0$$

What's the Point of Formulae?

- $2^n - 1$

What's the Point of Formulae?

- $2^n - 1$

Why is this important? What if we were writing code to solve this?