**Video 1 Features of a problem that can be solved using computational methods** [link](https://www.youtube.com/watch?v=r2c_SfdEQ84&list=PLCiOXwirraUAASApItjpR9v2QaTOAXMlS)

1. 0:30 is every problem computable?

2. 0:30 What did David Hilbert say in 1920?

3. 0:52 What did Alan Turing prove in 1936 with the Halting problem?

4. 0:52 What belief did Alan Turing’s Halting problem dispel?

5. 1:29 What 2 things can impose limitations on computable problems even if they are technically solvable?

6. 1:56 If a problem has an exponential complexity what does it require to solve it? In reality what does this actually mean?

7. 2:05 Does taking this action make any difference to solving the problem?

8. 2:15 When solving problems what are the key success criteria that we are looking for?

9. 2:33 What is a tractable problem?

10. 2:48 What is an intractable problem?

11. Quickly research the word heuristic and what it means

12. 3:11 How can using heuristic methods make a problem solvable?

13. 3:42 What are the 7 features of a computable problem

14. 4:13 Why does applying abstraction and decomposition to a problem make it easier to solve ?

15 4:29 How can enumeration help developers to solve problems?

16. 4:59 If a problem can be boiled down to pure theory then how can this problem be solved? Why are computers great at solving these types of problems?

17 5:10 What is a simulation? And what are its aims? What is automation? Which computational thinking techniques use in order to make the problems that they easier to solve?

18 5:55 If you wish to learn more about Turing’s halting problem watch the rest of the video and summarize below

**Video 2 Problem recognition** [**link**](https://www.youtube.com/watch?v=2zNqU7bC3V8&list=PLCiOXwirraUAASApItjpR9v2QaTOAXMlS&index=2)

1.0:19 problem recognition what is it?

2.Why do you need to be cautious and careful when defining a problem?

3.1:40 With regards to the traffic light queue issue what are the possible solution/s?

4.1:40 Is there only ever one solution?

5.1:54 When you are presented with problems in the exam what questions do you need to pose to yourself?

6. 2:05 When you have answered the questions about the problem in the exam what question do you need to ask yourself next?

7: 2:20 What do we mean by problem recognition? And what skills do we use during problem recognition?

**Video 3 Problem decomposition** [**link**](https://www.youtube.com/watch?v=RUusIWfNCak&list=PLCiOXwirraUAASApItjpR9v2QaTOAXMlS&index=3)

1 0:22 What is problem decomposition ?

2 0:33 For larger problems do you need to repeat the process? If so at what point should you stop?

3. 0:43 By decomposing the problem effectively what should be the next step in the process?

4. 0:54 This process described so far is called what “A T................approach”

5.1:02 Which type of problems are harder to break down in this way?

6.1:11 In an event driven problem how could you breakdown the problems ?

7.1:22 what is stepwise refinement?

8 2:16 On the example top down break down of the game what do the different coloured boxes/ layers represent?

9. 2:30 Is there a set way of completing these diagrams?

10. 2:39 What do we mean by problem decomposition?

**Video 4 Use of divide and conquer** [**link**](https://www.youtube.com/watch?v=CPrG77GOXAM&list=PLCiOXwirraUAASApItjpR9v2QaTOAXMlS&index=4)

1.0:18 What is the definition of divide and conquer?

2 0:32 What are the 4 general steps involved in this process?

3 0:39 Which is one of the best examples of this process?(we will cover this later next half term)

4 2:12 Using the police search example from the video describe how they use divide and conquer on the interface to reduce the number of records that they find

5 3:13 Which other computational method is divide and conquer similar to?

6 3:24 What else can divide and conquer help us to identify tasks that ….....

7 3:31 What other computational thinking skills is divide and conquer similar?

**Video 5 The use of Abstraction** [**link**](https://www.youtube.com/watch?v=xNHqa5oGCnM&list=PLCiOXwirraUAASApItjpR9v2QaTOAXMlS&index=5)

1. 0:18 What is abstraction

2. 0:43 When tackling a problem what is it important to consider?

3. 1:02 When tackling a problem what should you actively avoid in your potential solution?

4. 1:22 What type of software presents excellent examples of abstraction? Describe an example and how they use abstraction

5. 1:53 How can Abstraction be used to solve a problem?

**Video 6 Back tracking, data mining and heuristics** [**link**](https://www.youtube.com/watch?v=VgQGPn-2dG8&list=PLCiOXwirraUAASApItjpR9v2QaTOAXMlS&index=6)

1.0:43 What is the process of back tracking?

2. 1:01 What are the limitations of backtracking?

3. 1:09 What types of problem are best suited to backtracking?

4. 1:35 What

**Video 7 Applying performance modelling, pipelining and visualization to solve problems** [**link**](https://www.youtube.com/watch?v=37P9f3QTP58&list=PLCiOXwirraUAASApItjpR9v2QaTOAXMlS&index=7)