St Bede's Catholic College

Year 11 into 12 Transition Work

Computer Science

Exam board: OCR

Course length: Two years

Specification: https://www.ocr.org.uk/Images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf

Exam structure:

Content Overview	Assessment Overview	
• The characteristics of contemporary processors, input, output and storage devices • Software and software development • Exchanging data • Data types, data structures and algorithms • Legal, moral, cultural and ethical issues	Computer systems (01) 140 marks 2 hours and 30 minutes written paper (no calculators allowed)	**40%** of total A level
• Elements of computational thinking • Problem solving and programming • Algorithms to solve problems and standard algorithms	Algorithms and programming (02*) 140 marks 2 hours and 30 minutes written paper (no calculators allowed)	**40%** of total A level
The learner will choose a computing problem to work through according to the guidance in the specification. • Analysis of the problem • Design of the solution • Developing the solution • Evaluation	Programming project 03* – Repository or 04* – Postal or 80 – Carry forward (2018 onwards)* 70 marks Non-exam assessment	**20%** of total A level

Useful textbooks:

OCR AS and A Level Computer Science

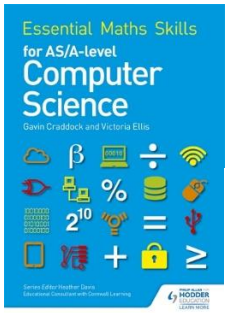
Author: PG Online Ltd

ISBN: 978-1-910523-05-6

Publisher: PG Online Ltd

Date: September 2016

This is a complete course text which includes AS and A Level for the H046 and H446 specifications. The book is divided into 12 sections, each containing roughly six chapters. Each chapter covers material that can comfortably be taught in one or two lessons. It will also be a useful reference and revision guide for students throughout the A Level course. Content applicable to the A Level only is indicated throughout the textbook.



ISBN: 9781471863578
Published: 27/05/2016
Extent: 104 pages

If you struggle with binary multiplication, or Big O Notation, this is the book for you. This textbook companion will help improve your essential maths skills for computer science, whichever awarding body specification you're following. You can use it throughout your course, whenever you feel you need some extra help.

- Develop your understanding of both maths and computer science with all worked examples and questions within a computer science context
- Improve your confidence with a step-by-step approach to every maths skill
- Measure your progress with guided and non-guided questions to see how you're improving
- Understand where you're going wrong with full worked solutions to every question
- Feel confident in expert guidance from experienced teachers and examiners Victoria Ellis and Gavin Craddock, reviewed by Dr Kathleen Maitland, Senior Lecturer in Computing and Director of the SAS Student Academy at Birmingham City University

Useful websites:

https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/

Sample/past papers:

https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/assessment/

Transition work:

Wider computing issues and integrated questions

These questions require you to use your technical knowledge in context. Reference any sources that you use to help you.

1. Create a timeline showing the history of computing, including any key discoveries or inventions. Extend you timeline to show how you think computer science might develop over the next 50 years.

2. Compare the Xbox ONE, PS4 Pro and PC as gaming platforms. You must use as much technical detail as possible and reference any evidence presented. Choose how you will present your ideas.

3. Discuss the benefits and limitations of Virtual Reality
    a. In business contexts, such as medicine
    b. As a gaming tool
    c. As an extension to social media

Systems Architecture

1. Produce an annotated diagram showing how the CPU processes data. This should include
    a. The purpose of the CPU
    b. Common CPU components and their function
        i. Arithmetic and Logic Unit (ALU)
        ii. Control Unit (CU)
        iii. Cache
        iv. Registers
            1. Memory Address Register (MAR)
            2. Memory Data Register (MDR)
            3. Program Counter
            4. Accumulator

    c. Reference to the fetch-execute cycle
2. Discuss, with examples, how the performance of a CPU can be improved, including:
    a. Increasing the clock speed
    b. Increasing the cache size
    c. Increasing the number of processing cores

Memory

1. Compare RAM and ROM
2. Explain the need for virtual memory in a computer system
3. Describe the characteristics of flash memory

Storage

1. Complete the following table comparing optical, magnetic and solid state storage media

	Capacity	Speed	Portability	Durability	Reliability	Cost
Optical						
Magnetic						
Solid State						

2. Justify one use of each storage method

Networks

1. Explain the similarities and differences between
    a. A LAN and a WAN
    b. Client-server and peer-to-peer networks
2. Explain the difference between the Internet and the World Wide Web
3. Describe the factors that affect network performance, and explain how network performance can be improved
4. Draw three different network topologies
    a. Label all the components required to create each network
    b. Explain the purpose of each component in the network, including
        i. Wireless Access Points
        ii. Routers
        iii. Switches
        iv. Network Interface Cards
        v. Transmission media, such as Ethernet Cables

5. There have been many recent high-profile cyber-attacks across the world, including the attack on the NHS in May 2017. Some commentators have said that "we now rely too much on technology". Write an essay explaining how far you agree with this statement and including descriptions of threats to IT systems and ways to reduce vulnerabilities.

Systems Software

Create a presentation comparing Windows, Linux, iOS, Android (which is based on Linux) and Unix. Discuss the features of each operating system, comparing the benefits and limitations of each. Note that you can try a basic Unix interface here: http://www.masswerk.at/jsuix/

Ethical, Legal, Cultural and Environmental Concerns

Find a recent news story on one of the following topics:

- A legal issue in computing, such as a breach of the Data Protection Act
- An ethical issue in computing, such as the development of AI
- An environmental issue in computing, such as the disposal of waste equipment
- A technical development in computer science, such as the Internet of Things

Summarise the story, explaining any technical content for a student in year 10. Explain how the story affects you as a student of computer science.

Computational Thinking – Theory

Computational Logic and Calculations

1. Complete the truth tables for the following expressions
    a. A AND (B OR C)

A	B	C	B OR C	A AND (B OR C)
0	0			
0	0			
0	1			
0	1			
1	0			
1	0			
1	1			
1	1			

b. (NOT A) OR (NOT B)

A	B			

a. What single logic gate produces the same result as the expression in part b?
b. Draw a circuit to represent each expression

2. Calculate each of the following, showing any appropriate working you need
   a. 13 MOD 2
   b. 16 MOD 6
   c. 15 MOD 3
   d. 7 MOD 8
   e. 13 DIV 2
   f. 16 DIV 6
   g. 15 DIV 3
   h. 7 DIV 8
   i. 2^0
   j. 2^7
   k. 2^8
   l. 2^10

3. Covert the following into the units given
   a. 4 bytes = bits
   b. 1 TB = bytes
   c. 80 kB = GB
   d. 40 MB = nibbles

4. Complete the table, converting between binary, hexadecimal and denary as required

Binary	Hex	Denary
0010 1010		
	0B	
		255
0110 0111		
	F5	

		48
	CD	

5. Complete the following calculations
    a. 0110 0011 + 0011 0001
    b. 1010 0110 + 1100 1111
    c. 0110 0011 << 2 (bit shift left two places)

6. Check if these are valid ASCII characters. If they are, give their character equivalent. Note that the first bit is a check digit using even parity, and the remaining 7 bits are the character a. 1110 0010
    b. 1100 0111
    c. 0011 0110
    d. 1100 1010

Programming Tools and Standards

1. Compare the use of jpg, png and gif to store images, explaining the benefits, properties and uses of each image format.

2. Produce an annotated diagram of the IDE you prefer to use to write code, explaining any features of the IDE that help you to produce your code. You may need to show examples of the IDE in use to highlight the different features. Tip: In school you have primarily used IDLE as your IDE for Python programming.

Extension Work (Optional)

Coding challenges

The coding challenges below will let you check your skills. Part of the transition to A-level is combining skills, and also ensuring that you plan and test your work thoroughly, so think about how you can re-use components and design your code for readability and robustness.

1. Write a program to:
    a. Ask the user to input
        i. Their first name
        ii. Their surname
        iii. A date, in the format DD/MM/YYYY
    b. The program should then output a customer ID as follows:
        i. The date in the format YYYYMMDD, then the first three letters of the surname, then the first initial, then the length of their first name. All letters should be in capitals
        ii. For example, John Smith, 27/05/2017 would give 20170527SMITHJ4
    c. The program should validate any inputs and keep asking for inputs until the user enters correct details or types "quit" at any point

    Plan your algorithm first, using a flowchart or pseudocode.

    Code your algorithm and provide evidence of both your code and the working output.

    Create a test plan for your algorithm, including testing your validation with normal, boundary and erroneous data.

2. Write a program to:
    a. Ask the user to input
        i. The name of a product
        ii. Its cost in pounds
        iii. The program should keep asking for inputs until the user types "None"
    b. The program should then output:
        i. The name and price of the most expensive item
        ii. The name and price of the least expensive item
        iii. The average price of the items
        iv. The total cost of the items
        v. Items over £50 get a 5% discount
        vi. VAT is added at the end at 20%
    c. The program should validate any inputs

    Plan your algorithm first, using a flowchart or pseudocode.

Code your algorithm, and provide evidence of both your code and the working output

Create a test plan for your algorithm, including testing your validation with normal, boundary and erroneous data.

Algorithmic Thinking and Problem Solving

The following puzzles will help you to develop you logical thinking skills. There are many good books of puzzles, plus countless online sources to test your skills. Some recommendations are given later.

The following puzzles are representative of classical problems and problem solving strategies. You can solve each one by trial and error, but you encouraged to think about the strategy you employed to solve the problem. Note that there are discussions of each problem available online if you want to investigate them in more detail.

Two good general strategies to try are:

- Can you solve a simpler version of the problem first?
- Can you draw a diagram to help you *visualise* the problem?

After that, you have your standard computer science strategies:

- Decomposition - Can you split the problem down into smaller parts to solve?
- Abstraction - Can you remove any unnecessary details to focus in on only what you need to solve the problem?
    - Be careful – are you sure that you have kept the right information?
- Generalisation and problem recognition - Is this puzzle a specific example of a problem for which there is a general solution? If so, how does it apply in this case?
    - Do you recognise the problem from somewhere else, or is it similar to something else?
        - You may need to generalise the problem to identify the core features so that you can spot equivalent problems.

Another important strategy is to ensure that the problem is *well-defined*. This means that you know:

- The goal: what you are trying to achieve
- The givens: what you know at the start, or your starting conditions
- The resources: what you have available to solve the problem
- The constraints: any rules that limit your solution
- The ownership: who or what is carrying out each part of the solution

Sometimes just working through the problem definition carefully is enough to give the required insight.

The complete work on problem solving is Polya's "How To Solve It"; there are many sources for this online if you are interested.

The problems

The Princess in the Castle

A princess lives in a long corridor in a castle. The corridor has 17 rooms, numbered 1 to 17 inclusive. Each night the princess sleeps in a different room according to the following rules:

- On the first night of the year she sleeps in a random room
- Each night she moves to an adjacent room; she never sleeps in the same room on two nights in a row and she always moves exactly one room left or right along the corridor. For example, if she is currently sleeping in room 12, then on the next night she will either be in room 11 or in room 13
    - If she is in room 1, then she must be in room 2 on the next night as she cannot move in any other direction (the same is true for room 17 – she must move to room 16 next)

A prince wishes to marry the princess. To do this he must find her room in the castle. However, whenever he sneaks into the castle at night, the guards quickly find him and throw him out! Therefore, he only has time to search one room each night.

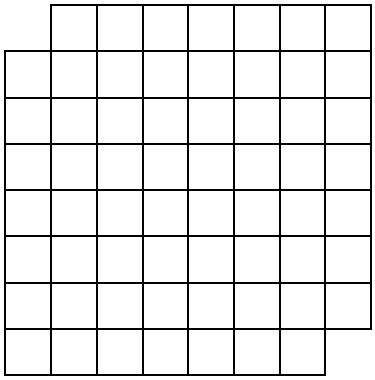
The princess is unable to give the prince any clues to her location, and the prince has no knowledge of her location, other than whether or no she was in the room he last tried.

What strategy should the prince follow in order to find the princess in a finite time?

What is the maximum number of nights the prince needs to search before he can guarantee finding the princess?

Covering a Chess Board

Imagine a standard 8x8 chess board. Now cut off two diagonally opposite corners squares to get a shape like this:



I also have a pack of dominoes. Each domino is exactly the right size to cover two squares on the chess board, either horizontally or vertically. (The dominoes cannot be placed diagonally.)

Is it possible to cover the board with dominoes so that each domino covers exactly two squares, with no overlaps and without any dominoes "hanging off" the edge of the board? If so, how do you do it? If not, why not?

Hat, hat, hat…

I have taken a group of students on a school trip. I want to organise them into two groups and so I have given each one a coloured hat. Some hats are red, while others are blue. Each child can see everyone else's hats, but not their own.

I have asked the students to get themselves into two groups based on the colours of their hats, with all the red hats together and all the blue hats together. But! I have told them they are not allowed to talk or communicate in any way.

What strategy should they use to form the two groups?

Einstein's riddle (and related grid problems)

Grid puzzles have been in print for years.

	Python	Java	VB	C	Puzzles	Maths	Gaming	Money
Alice								
Bob								
Charlie								
Dave								
Puzzles								
Maths								
Gaming								
Money								

1. Of the one who likes puzzles and the one who loves maths, one is Alice and the other programs in C.
2. The python programmer's name is alphabetically one more than the person who enjoys solving puzzles
3. Bob got into computer science through gaming
4. Of Dave and Bob, one wants to study computer science for the money, while the other codes in VB

The torch and the bridge

Three travellers wish to cross a rickety old rope bridge. Each person takes a different amount of time to cross the bridge.

- Alice takes 1 minute
- Bob takes 2 minutes
- Charlie takes 5 minutes
- Dave takes 8 minutes

The bridge will only support two people at once (it is very old)

What's worse, we only have one torch between us… It is (of course) a very dark night and the bridge is too dangerous to cross without the torch. Oh, and the torch only has enough battery for 15 minutes…

How do we get across the bridge?

Light switches

You are standing in a room with three light switches. Each switch controls exactly one light bulb in the next room. (This is a budget puzzle, so they are plain, cheap, basic light bulbs.) The door to the next room is closed, and there are no windows, so you cannot see the light bulbs.

You may manipulate the switches as much as you like, then you may go through into the room with the lights. You must then say which switch controls which bulb.

How do you do it?

Knights, knaves and spies

On the fabled Island of Knights and Knaves, we meet three people, A, B, and C, one of whom is a knight, one a knave, and one a spy. The knight always tells the truth, the knave always lies, and the spy can either lie or tell the truth.

A says: "C is a knave."
B says: "A is a knight."
C says: "I am the spy."

Who is the knight, who the knave, and who the spy?

Weighing and measuring

1. You have 10 bags of coins; each bag contains 100 coins. Nine of the bags contain real coins; each real coin weighs 1 gram. One bag contains fake coins; each fake coin weighs 0.9 grams.

   If you have an accurate scale that will display the weight of an object placed on it, how can you identify the bag of forgeries using the scale only once?

2. You have 12 coins, one of which is fake. The fake is either lighter or heavier than the real coins, but you do not know which. You have a balance that you can use to compare the weights of items.

   How can you find the fake coin in just three uses of the balance? (You have no other weights or reference objects, just the balance and 12 coins.

Make 15

You and I are going to play a card game. The rules are as follows:

- 9 cards, numbered 1 – 9, a placed face up on the table between us
- You go first

- On your turn you may pick up any one card from the table
- We alternate turns, each picking up one card at a time
- The winner is the first player to get any three cards that add up to exactly 15 (You can have more than three cards in your hand as long as three of them add up to 15. For example, if I was holding 8, 6, 2 and I could pick up the 5 I would win with 8, 2, 5)

What strategy should you follow to always win at this game, or at least never lose?

Mr Hirani is Head of Computer Science. Please email him on m.hirani@stbcc.org with any queries.