Get Ready for Computer Science A-Level

Subject mind-set and BIG picture thinking & links…

Computer Science is the science of problem solving. In Computer Science we look for ways to methodically solve problems.

A mathematician called Edsger Dijkstra said "Computer science is no more about computers than astronomy is about telescopes."

What he meant was that telescopes are tools used by astronomers to study the universe, and that computers are tools used by computer scientists to solve problems. It is important for Computer Scientists to understand how computers work, just as it is important for astronomers to know about telescopes but is important to remember that they are tools and not an end in themselves.

Watch this video.

https://youtu.be/AnVBCyzNgaU

You should take every opportunity to read, watch or listen to articles about technology and news stories about technology. In the A-Level course there are topics about ethical, social and legal issues relating to technology. You should be aware of issues and also keep up to date with news stories about technology.

Please explore the following tasks:

Work to complete

It will not surprise you when I tell you that programming is a key part of the course. Programming is a fundamental way in which we use computers to solve problems.

Warm up by doing these (quite simple) pseudocode challenges.

Pseudocode Challenge website

http://tools.withcode.uk/ks5pseudo/

Since people reading this will have varied amounts of experience in programming, I suggest that you select the tasks that are appropriate to your level of experience.

Watch this introduction to Python:

https://www.youtube.com/watch?v=kqtD5dpn9C8&list=RDCMUCWv7vMbMWH4-V0ZXdmDpPBA&start_radio=1&rv=kqtD5dpn9C8&t=716

"Python for Beginners - Learn Python in 1 Hour"

(Programming with Mosh)

Watch the video and tackle the activities as you go along. Pause the video as required. Study all the topics as per list below:

0:00:00 Introduction

0:00:30 What You Can Do With Python

0:01:15 Your First Python Program

0:05:30 Variables

0:09:08 Receiving Input

0:10:48 Type Conversion

0:18:49 Strings

0:23:41 Arithmetic Operators

0:25:59 Operator Precedence

0:27:11 Comparison Operators

0:28:52 Logical Operators

0:31:06 If Statements

0:36:16 Exercise

0:41:42 While Loops

0:45:11 Lists

0:48:47 List Methods

0:52:16 For Loops

0:54:54 The range() Function

0:57:43 Tuples

Once done, complete the python programming task enclosed.

Once you have done the task to a basic level, you should improve your solution by:

Adding appropriate validation to ensure that your program can't be crashed or made to produce incorrect output by erroneous data being entered.

Make your program as easy to read and as easy to maintain as possible by:

Adding comments to your code

Writing your code in a structured way using procedures and functions where appropriate.

Programming Practice Continued:

Go to the w3schools.com website using the following link.
https://www.w3schools.com/python/python_intro.asp

Register in the W3 schools website using your email address.

The following sections should be worked through by coding along.

This should give you a good grounding for the Y12 A Level.

- Python - Introduction
- Python - Variables
- Python - Data Types, Numbers & Casting
- Python - Strings
- Python - Lists
- Python - Dictionaries
- Python - Conditionals, Loops, Iterators, Polymorphism
- Python - Arrays, Objects, Inheritance & Scope
- Python - Functions, Lambdas
- Python - File Handling

Recommended textbooks

Course Book – "OCR AS and A Level Computer Science" – Heathcote & Heathcote (PG Online)

https://www.pgonline.co.uk/resources/computer-science/a-level-ocr/ocr-a-level-textbook/

This is the official course textbook, reading some of this before the course starts will help you hit the ground running.

Recommended websites you should be familiar with

Craig and Dave are Computer Science Teachers. They have made videos that address just about every aspect of the course. This is their website. They also have put most of their videos on YouTube. Please remember that we are doing the OCR specification.
https://student.craigndave.org/

Isaac Computer Science is a website with lessons for A-Level Computer Science students.
https://isaaccomputerscience.org/topics/a_level#ocr

The Computerphile channel on YouTube has many excellent videos. Here is Tome Scott explaining why electronic voting is, in his opinion, a very bad idea.

https://youtu.be/w3_0x6oaDmI

Here are some other sites with course materials put on-line as recommended by OCR.

Carnegie Mellon University – Principles of Computing

Stanford University – Coursera, Computer Science 101

Related magazines, blogs subject experts

Component 1 includes study of the "moral, social, ethical and cultural opportunities and risks of digital technology". Think about how AI is changing society.

Article: "Fooling YouTube's copyright-detecting AI"

www.theregister.co.uk/2019/06/20/avoid_youtube_copyright_ai/

Are we making the world unsafe? Article: "IoT security: Why it will get worse before it gets better"

https://zd.net/2BpFEK1

Component 1 covers Cybersecurity: "Network security and threats, use of firewalls, proxies and encryption." – are we making the world unsafe?

OCR Learner Activity "Thinking Logically"

www.ocr.org.uk/Images/253773-thinking-logically-activity.docx

Recommended books & revision guides

See above under "Recommended Textbooks".

See also this book - free on Wikibooks (but no substitute for the recommended text book)
A-level Computing, by Peter Kemp, available on WikiBooks: Wikibooks – OCR A-level Computing

Possible virtual places to visit

Bletchley Park, where Alan Turing and others worked to decrypt encrypted enemy messages (encrypted using The Enigma Machine) during World War II.
https://bletchleypark.org.uk/

Teachers contact for questions and more information
Mr Mehdi (Salah.Mehdi@shoreham-academy.org)

It is time to look forward and explore the courses you planning to study in the future.

Getting organised: Begin to set up a Learning folder digital or at home for your chosen subjects.

(Shoreham Sixth Form has a way of organising this)

OCR
Oxford Cambridge and RSA

Released June 2015
For Assessment Submission
June 2017

GCSE COMPUTING

A453   Programming Project

CONTROLLED ASSESSMENT MATERIAL 3

This assessment may be periodically reviewed. Please check on OCR Interchange that you have the Controlled Assessment material valid for the appropriate assessment session.
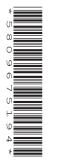
INSTRUCTIONS TO TEACHERS

*   Please refer to Section 4 of the GCSE Computing specification for instructions on completing this controlled assessment task.
*   The marking criteria should be available to candidates whilst completing the tasks.
*   The quality of written communication will be assessed in the testing section.
*   The total number of marks for this unit is **45**.

INFORMATION FOR CANDIDATES

*   This document consists of **4** pages. Any blank pages are indicated.

Teachers are responsible for ensuring that assessment is carried out against the Controlled Assessment set for the relevant examination series (detailed above).

Assessment evidence produced that does not reflect the relevant examination series will not be accepted.

This assessment consists of three tasks.

Candidates should complete all tasks and provide evidence to meet all the marking criteria.

For the following scenario analyse the detailed requirements for each situation and, using suitable algorithms, design a solution to be coded in a suitable high-level programming language. Show the iterative development of the individual solutions with suitable testing throughout the process. Test the final products and evaluate your solutions against the detailed requirements you identified in the analysis.

The results for a task may be used without further testing in any subsequent task, or each of the tasks may be solved as a separate system.

Troubleshooting system

Task 1

When a customer has a problem with a product, automated troubleshooting programs are available to help them.

If a problem with a mobile device occurs, then a troubleshooting process is invoked, with the user responding to a series of questions that lead to a solution or advice to contact the supplier directly.

For example, a conversation might proceed as follows:

System: 'Has the phone got wet?'
User: 'No'
System: 'Has the phone been dropped?'
User: 'No'
System: 'Is the phone fully charged?'
…

Not all of the questions need to have yes or no responses but each response will need to lead to a supplementary question or a solution.

Analyse the requirements for this system and design, develop, test and evaluate a program to interrogate the user about the nature of the device and the problem, leading to a solution or advice to call the supplier. You will need to create a suitable troubleshooting tree for a mobile device. This should be a demonstration version of the system with at least 10 possible outcomes, but need not include an exhaustive list of potential problems or outcomes.

Task 2

Some automated troubleshooting programs identify certain keywords in the customer's query to provide potential solutions to the query.

For example, 'There is no display on my mobile phone screen.'

The keywords 'display' and 'phone' would link to common problems with a phone display.

Analyse the requirements for this system and design, develop, test and evaluate a program to identify keywords in a query typed in by the user and provide a linked solution to common problems related to a mobile device from a selection stored in a text file or database. You will need to identify appropriate keywords that can be linked to general advice related problems. You need to account for variations in the form of the user input e.g. one user may query 'There is no display on my mobile phone', another user may query 'My phone screen is blank.'

Task 3

Troubleshooting systems need to identify the device and load the correct troubleshooting routine before analysing the problem.

For example, initialising the troubleshooting routine may require this conversation:

System: 'What type of device?'
User: 'Phone'
System: 'Which make of phone is it?'
User: 'IPhone'
System: 'Which model of phone is it?'
User: 'IPhone5'
System: 'Which version, S or C?'
User: 'S'
System: 'How much memory, 8GB, 16GB or 32GB?'
User: '16'
System: 'What is the problem?'
User: 'There is no display!'
….

If no solution is found, the data is allocated a case number and stored to be sent to a technician.

Analyse the requirements for this system and design, develop, test and evaluate a program to provide troubleshooting advice for several devices. The system should identify the device and load a file of suitable questions and responses for that device. You will need to create two files to test the system, for third or subsequent devices it is acceptable to show that they have been recognised with a simple acknowledgement. This is a demonstration version of the system and need not deal with all potential problems.