OCR A-Level

Computer Science

H446

Bridging work

Introduction

We want you to be really successful and what it takes to be successful at GCSEs is different from being successful at A-Levels. Although you have fewer subjects, there are different skills at A-Level and the volume of work is greater because the depth and detail is more demanding.
The Bridging work will reassure you that the subject you have selected is for you. We want you to study a course that interests you and you are sufficiently qualified to study.

Aim of the Bridging work

To ensure secure foundations for starting A-Level Computer Science in September.

Course overview

The course specification can be viewed here

Aims and learning outcomes

The aims of this qualification are to enable learners to develop:

- an understanding of and ability to apply the fundamental principles and concepts of computer science including; abstraction, decomposition, logic, algorithms and data representation
- the ability to analyse problems in computational terms through practical experience of solving such problems including writing programs to do so
- the capacity for thinking creatively, innovatively, analytically, logically and critically
- the capacity to see relationships between different aspects of computer science
- mathematical skills
- the ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology.

Key features of the specification

The OCR A Level in Computer Science will encourage learners to be inspired, motivated and challenged by following a broad, coherent, practical, satisfying and worthwhile course of study. It will provide insight into, and experience of how computer science works, stimulating learners' curiosity and encouraging them to engage with computer science in their everyday lives and to make informed choices about further study or career choices.

The key features of this specification encourage:

- emphasis on problem solving using computers
- emphasis on computer programming and algorithms
- emphasis on the mathematical skills used to express computational laws and processes, e.g. Boolean algebra/logic and comparison of the complexity of algorithms
- less emphasis on ICT.

Centres and learners have the opportunity to:

- produce a programming project which is more refined and more focussed on coding
- choose the project title and problem to be solved
- choose any suitable programming language

<table>
<tr><th>Content Overview</th><th colspan="2">Assessment Overview</th></tr>
<tr><td rowspan="3">• 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

• Elements of computational thinking
• Problem solving and programming
• Algorithms to solve problems and standard algorithms

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</td><td>Computer systems
(01)
140 marks
2 hours and 30 minutes
written paper
(no calculators allowed)</td><td>40%
of total
A level</td></tr>
<tr><td>Algorithms and programming
(02*)
140 marks
2 hours and 30 minutes
written paper
(no calculators allowed)</td><td>40%
of total
A level</td></tr>
<tr><td>Programming project
03* – Repository
or
04* – Postal
or
80 – Carry forward
(2018 onwards)*
70 marks
Non-exam assessment</td><td>20%
of total
A level</td></tr>
</table>

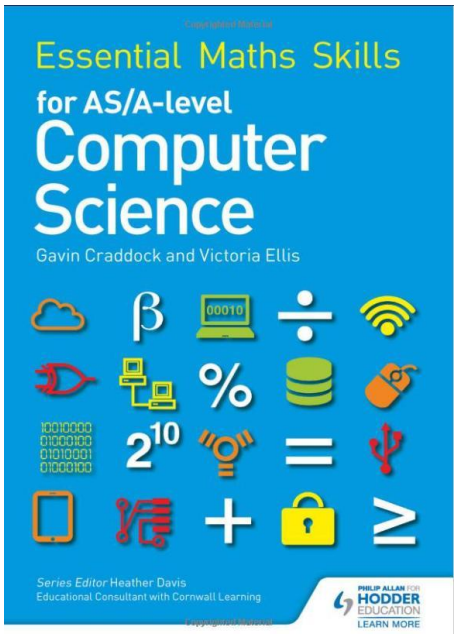
Textbooks needed for September

Main textbook (Mandatory)

Paperback: 378 pages
Publisher: PG Online Limited (12 Sept. 2016)
Language: English
ISBN-10: 1910523054
ISBN-13: 978-1910523056
Product Dimensions: 21.6 x 2.6 x 28 cm

±£32.00

Essential Maths skills for AS/A-Level Computer Science (Optional)



Paperback: 104 pages
Publisher: Philip Allan; UK ed. edition (27 May 2016)
Language: English
ISBN-10: 1471863573
ISBN-13: 978-1471863578
Product Dimensions: 21 x 1 x 29.4 cm

±£11.50

Tasks

Complete all the tasks and evidence it in a Word document (or similar), font size 12. Convert the document to a PDF (in MS Word you can Save As--->PDF) and email it to me:

Mr JJ Pieterse – jpieterse@kts.school

If you already have an official KTS email address (e.g. anything@kts.school), please email from that account, otherwise from your normal email account.

Deadline: Monday, 9 September 2024, by 9am

Programming

Over the summer you need to either maintain or develop a good level of programming. We mainly use Python in our day to day lessons, but we will be learning the foundations of C# (C sharp) in Y12.

When you start your programming project later in Y12, you can either use Python or C# to program your Graphical User Interface (GUI) project. You'll also have the option to use a different programming language of your choice if it's approved by OCR. ***The main issue is that whichever programming language you choose for your programming project, you must be very familiar with it by around June 2025!***

Please download and install the latest version of Visual Studio for your home computer before September 2022 here:

Visual Studio 2022 (Community edition)

If you don't have Python on your home computer, please download and install it from here:

Python 3.12.4 (Windows 10)

Python 3.12.4 (macOS)

Remember to choose the correct version for your hardware e.g. Windows Installer (64-bit)

Task 1

Computer Science is all about building systems and solving problems that have a direct influence on people's lives. As you start the course, write a single paragraph describing one area of Computer Science that you are particularly interested in, and one in which you could imagine yourself working in the future. Examples include Programming, Games Design, Problem Solving, Computer Graphics, Web Page building, Big Data, Machine Learning, Internet of Things, etc.

After completing the paragraph discuss why you would like to work in that area, what specifically interests you about it?

Minimum length: about half of A4

Task 2

It is important that you understand the subject related terminology if you are to be successful in this course.
The following are some of the key terms will need to be learnt in order for you to form a better understanding of Computing. Research a brief definition for each and add to your document.

Von Neuman Architecture
Assembly Language
Machine Code
Compiler
Object Oriented Programming
Lossy Compression
Hashing
Packet Switching
ASCII

Minimum length: about page/page and half of A4

Task 3

Coding is an important part of A Level Computer Science. Use the following resource to develop your knowledge and experience of the Python programming language:

https://www.learnpython.org/en/

Work through each section up to and including Functions.

Learn the Basics

- Hello, World!
- Variables and Types
- Lists
- Basic Operators
- String Formatting
- Basic String Operations
- Conditions
- Loops
- Functions
- Classes and Objects
- Dictionaries
- Modules and Packages

Minimum length: Paste a screenshot of one piece of code and the result of it for each of the 9 sections (Don't crop it too small or stretch it to wide/long)

Task 4

Programming Task: Please choose one of the following projects to complete:

Project Choice 1 (Medium) - Dice Rolling Simulator
As the name of the program suggests, we will be imitating a rolling dice. This is one of the interesting python projects and will generate a random number each dice the program runs, and the users can use the dice repeatedly for as long as he wants. When the user rolls the dice, the program will generate a random number between 1 and 6 (as on a standard dice).

The number will then be displayed to the user. It will also ask users if they would like to roll the dice again. The program should also include a function that can randomly grab a number within 1 to 6 and print it. This project will help build a strong foundation for fundamental programming concepts.

Project Choice 2 (Medium) - Number Guessing
Make a program in which the computer randomly chooses a number between 1 to 10, 1 to 100, or any range. Then give users a hint to guess the number. Every time the user guesses wrong, they get another clue, and their score gets reduced. The clue could be multiples, divisible, greater or smaller, or a combination of all.

Project Choice 3 (Harder) – Hangman Game
This is more of a "guess the word" game. The core concepts you have to use while developing this project are variables, random, integer, strings, char, input and output, and boolean. In the game, users have to enter letter guesses, and each user will have a limited number of guesses (a counter variable is needed for limiting the guesses).
You can create a pre-organised list of words that users can grab words from. You must include specific functions such as those to check whether or not a user has entered a single letter, if the input letter is in the hidden word, if the user has actually inputted a single letter, and to print the correct letters as they are guessed.

Minimum length: Paste screenshots of your code and one or two results of the working program

I look forward to receiving your quality work!

Something to watch to broaden your knowledge/scope of Computer Science

TED Talks:

This talk on the impact that artificial intelligence and robotics will have on jobs in the future.
The jobs we'll lose to machines - and the ones we won't (Length: 4:27)
TED Talk – Anthony Goldbloom

This talk on the internet of things
Everything around you can become a computer (Length: 13:02)
TED Talk – Ivan Poupyrev

This talk which explores some of the philosophical ideas surrounding artificial intelligence.
What happens when our computers are smarter than we are? (Length: 16:22)
TED Talk – Nick Bostrom

Films:

Hackers: The hero of the story is arrested for writing a computer virus as a teen, and as an adult later works with his friends to take down a plot to release a dangerous computer virus.

Wargames: Oops! An unsuspecting computer prodigy accidentally starts World War III when he hacks into a military computer to play a war game. Enjoy the ideas of computer control, automation, and the importance of being error free.

The Matrix: Most computer programmers live a pretty typical life, but Neo finds out that his reality is in fact not a reality at all, and rather a computerized artificial world. Neo offers inspiration to all who care about the importance of real human life over virtual reality.

Tron: Have you ever felt trapped by your computer? In this movie, a hacker actually is, and has to overcome a program that holds him captive. Students can enjoy this movie, and it's sequel Tron: Legacy for their lesson in being bigger than your computer program.

Antitrust: If you think you're set for life when you get hired for your programming dream job, think again. This movie takes a look at the potential for ruthlessness and danger in the computer science industry.

Ghost in the Shell: An animated movie from Japan, Ghost in the Shell tells the story of a well-connected future set in 2029 full of hackers and the cyborgs that prevent them. Ghost in the Shell will teach computer science students to be wary of over connectedness.

The Imitation Game: Alan Turing coding our way to victory in WWII