Year 12 Transitions Programme 2020
Computer Science

UTC SOUTH DURHAM
LEARN DESIGN MAKE

Course Overview

You have made an excellent choice in deciding to study Computer Science at A-Level!

This pack contains a programme of activities and resources to prepare you to start an A-Level in Computer Science in September. It is aimed to be used throughout the remainder of the summer term and over the summer holidays to ensure you are ready to start your course in September.

The pack is divided into some of the key topics you will study in A-Level Computer Science. There are a range of different activities to do in each topic area. The world of Computer Science continues to develop at an amazing rate. The challenge for you as a computer scientist is to be able to respond to this ever changing world and to develop the knowledge and skills that will help you to understand technology that hasn't yet been invented!

Computer science academic success will require students:

1. To have a technical interest in computing and computer programming.
2. To stay up to date with current technology news and developments in computing.
3. To develop your computer programming skills outside of the classroom to ensure you can program with confidence.
4. To engage in classroom discussion and contribute to the learning environment and use free independent study periods to review and recap your learning.
5. To be a problem solver - if you are stuck then make an effort to find out the answer for yourself. Use Google, it contains so many fantastic tutorials and guides to help you, or read a textbook/revision guide related to the topic.
6. To be proactive and actively look for ways to expand your knowledge and get better at each stage and topic.

Ofsted Good Provider

University of Sunderland

Gestamp

HITACHI

Course Overview

Computer Science is a practical subject where students can apply the academic principles learned in the classroom to real-world systems. The aims of this qualification are to enable learners to develop the following:

1. An understanding and ability to apply the fundamental principles and concepts of computer science, including: **computational thinking** (abstraction, decomposition, pattern recognition**) logic and Boolean logic, algorithms, data representation and object oriented programming (OOP).**

2. The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs (using python or java programming languages).

3. The capacity to think creatively, analytically, logically and critically.

4. Mathematical skills to understand data types and primitive data types. The ability to solve binary arithmetic and floating point arithmetic and solve logic problems using Boolean algebra.

The key features of this specification encourage the following:

1. Emphasis on problem solving using computers.

2. Emphasis on computer programming and algorithms.

3. Emphasis on the mathematical skills used to express computational laws and processes, e.g. Boolean algebra/logic and comparison of the complexity of algorithms.

4. Theory of computation.

5. Fundamentals of data representation.

6. Fundamentals of computer organization & architecture.
7. Systematic approach to problem solving.

8. Learn about data structures. Data in a computer program is organized using a data structure. There are different methods for organizing data. Arrays are a common tool used to organized data when programming.

9. Understand databases and how data is exchanged between different systems.

10. Legal, moral, cultural and ethical issues surrounding the use of computers and ethical issues that can or may in the future arise from the use of computers.

11. The use of algorithms to describe problems and the measures and methods to deter-mine the efficiency of different algorithms, Big O notation.

Develop and complete a programming project, choosing project title and problem to be solved and using the agile approach (*agile software development is a process for developing software, allowing you (the developer*) *to create your software program in the way most suited).*

Task One: Computational Thinking

This task should take you approximately 45 – 60 minutes.

1. **Read this page from Cambridge Maths.**

2. **Watch this video of a complete lesson on Algorithms from Harvard University.**

3. **Read the following:**

*So what actually is 'computational thinking'? You should have a basic understanding from your GCSE years***.** *It is the thought processes involved in problem solving, so that the solutions are represented in a form that can be effectively carried out by an information-processing agent, such as a computer. Core concepts involved in computational thinking include: algorithmic thinking - developing a set of instructions or sequence of steps to solve a problem; evaluation - ensuring a solution is fit-for-purpose; decomposition - breaking a problem down into its component parts; abstraction - hiding detail or removing complexity without losing the important detail required to solve a problem; generalisation - finding a general approach to a set of problems.*

Creativity is important when applying computational thinking principles to a problem. Programming is a fundamentally creative skill - whether it is used to create a search algorithm, build an app or design a website.

Why is it important in the workplace? So much of modern day business is about problem solving - whether that's making small improvements to enhance the efficiency of a business, or creating breakthrough products and services for consumers. Computational thinking runs through every aspect and function of a modern business. It has become more crucial in the 21st century workplace where so much is now data-driven - analysing consumer behaviour, the movement in financial markets and the performance of public services, like health or policing, are just a few job roles that require individuals to be able to think through problems in a way that a computer could understand.

4. Answer the following using complete sentences and with examples:

a) What is an algorithm?
b) What is computational thinking?
c) Why is computational thinking required in the study of computer science?
d) What is abstraction and provide an example.
e) What is decomposition and provide an example.

Task Two: Artificial Intelligence and Recent Developments

BBC Click

BBC Click is the programme for everyone interested in the internet and computing. Whether it's e-commerce, new developments and products, or gadgets and games, BBC Click looks at the tools that will revolutionise business and personal life in the future.

This is a really useful website for staying up to date with current technological news and sup-port you in the A level topic concerning ethics, morals, legal and cultural issues surrounding computer science.

1. **Watch the following:**

BBC Click Review of 2019

This includes space travel, electric cars, 5G and the increased use by police of facial recognition.

Artificial Intelligence (AI) and Coronavirus

The following video looks at how AI tools could help track the virus spread and find new treatments.

- You will learn about AI at A Level and will be required to answer an essay question in the exam that may relate to AI.
- Do some further research on AI and see below example exam question and have a go at answering.

2. Answer the following question:

"Developments in Artificial Intelligence mean that in twenty years time most people will be unemployed." Discuss whether or not you agree with this statement.

Task Three: Understanding Technology

You will discuss and learn about systems security, firewalls and encryption during your A level.

1. Watch this video on protecting your data from Harvard University.

2. Watch/listen to this Ted Talk: “Why study computer science!”

3. Answer the following:

a) Why have you chosen the study Computer Science at A Level?

b) What do you aim to achieve from the course?

c) What skills are you aiming to develop?

d) What other A Levels have you chosen and why?

e) What programming languages have you used?

Task Four: Data Representation

During your GCSE you learnt about binary, denary, hexadecimal, binary arithmetic and ASCII. For A-Level you will also learn about the following;

- representing negative numbers using two’s complement
- fixed point binary numbers
- floating point arithmetic

Complete the following to help remind you of the skills from GCSE or re-learn the basic data representation techniques.

1. Read the BBC bitesize data representation pages (10 pages)
2. Complete the quiz and make a note of your score:
3. Read this on data representation.
4. Complete the data representation exam questions by following this link. Always show your working.

Task Five: Systems Architecture

One of the first topics you will start at A-Level is about the components of a computer. The CPU, also known simply as the processor, has a number of different components which enable it to carry out its task of executing instructions. These components include:

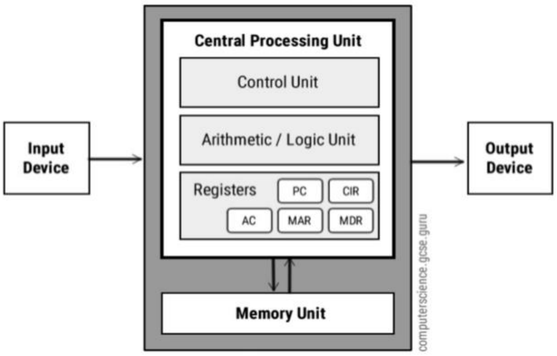
- Control unit (CU)
- System Busses (Control, Data & Address bus)
- ALU (Arithmetic Logic Unit)
- Dedicated registers
- Program counter (PC)
- Current instruction register (CIR)
- Memory address register (MAR)
- Memory data register (MDR)
- Accumulator (ACC)

The sequencing of these instructions can be divided into 3 phrases using the:

Fetch—Decode—Execute cycle.

1. Read about the history of computing. Choosing at least 6 historic computing moments, create a time-line explaining each of these and the impact it has had on computing today.

2. Research each of the keywords above (components and registers) and create a revision guide which explains each element. You can use the diagram (or create your own) below to complete this task. Watch this video.



computerscience.gcse.guru

Task Six: Python Programming and OOP (Object Oriented Programming)

What is Procedural programming (Python) vs OOP (Object Oriented Programming)

Procedural Programming can be defined as a programming model which is derived from structured programming, based upon the concept of calling procedure. Procedures, also known as routines, subroutines or functions, simply consist of a series of computational steps to be carried out. During a program's execution, any given procedure might be called at any point, including by other procedures or itself.

Object Oriented Programming can be defined as a programming model which is based upon the concept of objects. Objects contain data in the form of attributes and code in the form of methods. In object oriented programming, computer programs are designed using the concept of objects that interact with real world.

Object oriented programming languages are various but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

You will need to be programming regularly during this course (i.e. weekly) during your own time and during free study periods.

1. **Complete the Python programming course**

Python programming (or Java if you would prefer to use) is fundamental for A-Level computing. You will be required to undertake the project using a programming language and this will require a good understanding and ability to program and problem solve your programs successfully.

Complete this python programming course—this will take you approximately 10-15 hours. You will get a certificate at the end which you can share with your teacher.

2. **Complete the OOP (Object oriented programming) course**

Complete the following OOP course, which will take you 10-20 hours. OOP is fundamental to A-Level and will help with your programming project. OOP is how the world is viewed as a col-lection of objects. An object might be a person, car or animal. You can then make 'blueprints' with that object and re-use its attributes and behaviours.

Find out more about OOP by watching this video

Complete this OOP course – you will get a certificate at the end of the course to share with your teacher.

Once everything has been completed, please e-mail the work to gbenga.aleshe@utcsouthdurham.org, including your name, current school and a little background about yourself.