Computer Science: A-Level

Course Overview

- **Exam Board** : AQA
- **Usual Age Range** : 16-19
- **Qualification :** 1 A-Level
- **Curriculum Time** : Five 50 minute lessons per week in class plus additional work in Independent Learning Time
- **Assessment** : this curriculum is assessed via:
    - 2 x 2 hour 30 minute exams
    - 1 x Non-Examined extended project
- **Grading** – A* - U
- **Full specification** - AS and A-level Computer Science Specification Specifications for first teaching in 2015 (aqa.org.uk)

Curriculum Intent

The **intent** of the Computer Science curriculum is to give UTC students an opportunity to develop their understanding of the fundamental principles and concepts of Computer Science along with practical programming skills. The intent is to ensure students have advanced theoretical and practical knowledge, understanding and skills that can be applied in any Computing setting in their future career and of particular use to students considering a career in computing.

The further intent of the Curriculum is to give students useful technical skills around advanced programming such as the ability to write an effective flow chart, produce Pseudocode, and develop an understanding for advanced programming skills such as Object Orientated Programming and Unity Development.

Students are supported and encouraged to develop their **love of reading** and literacy skills on this course, by reading related computing news and articles and by completing regular extended writing activities.

Students are encouraged to develop their **numeracy** on this course by applying the mathematical skills relevant to Computer Science: including number systems of binary and hexadecimal; binary additional as well as calculating file sizes for images and sound.

Suggested next step **destinations** after completion include degrees in Computer Science, apprenticeships and employment in the emerging digital economy.

Related **careers** include working as a software developer; network support; games developer; cyber security specialist; systems analysis. This intent of the Curriculum is to also provide a good baseline knowledge, skills and understanding for students who undertake an Apprenticeship.

Remote Learning and Revision

Students will benefit additional study of A level theory content for the exam revision and also if they are absent from the UTC but well enough to complete remote learning. Students can communicate with the teacher via Teams or via email if absent from school.

- Isaac Computer – Isaac Computer Science
- Test and Track – Test&Track (testandtrack.io)
- Begginners Programming in Python – https://www.tutorialspoint.com/python/index.htm
- CraignDave: https://student.craigndave.org/7516-7517
- Practice Assessments and papers - https://tinyurl.com/2p876te6
- Topic and module quizzes: https://quizizz.com/admin
- ZigZag revision materials – available on Google Classroom
- ZigZag topic tests – available on Google Classroom
- Students can access all lesson and revision materials on Google Classroom

Curriculum Overview

The learning in A-Level Computer Science is sequenced as follows.

Note: the full Curriculum Plans are available on request to info@nef.tynecoast.academy

Key Topics

- Fundamentals of Programming
- Fundamentals of Data Structures
- Fundamentals of Algorithms
- Theory of Computation
- Fundamentals of Data Representation
- Fundamentals of Computer Systems
- Fundamentals of Computer Organisation and Architecture
- Consequences of Uses of Computing
- Fundamentals of Communication and Networking
- Fundamentals of Databases
- Big Data
- Fundamentals of Functional Programming
- Systematic Approach to Problem Solving
- Non-exam assessment – the computing practical project.

Year 12:

Half Term 1

- Fundamentals of Programming
- Using Variables, Selection, Looping and Arrays
- Fundamentals of Data Representation

Half Term 2

- Fundamentals of Computer Systems: including Hardware and Software; Languages and Translators; Boolean Algebra
- Sub procedures and Functions, String Handling, Validation and Exception Handling

Half Term 3

- Fundamentals of Data Structures
- Text File Handling, Classes and OOP
- Fundamentals of Computer Organisation and Architecture

Half term 4

- Fundamentals of Computer Organisation and Architecture
- Introduction to Pre-release code activities

Half Term 5

- Introduction to Non Exam Assessment and Analysis
- Consequences of Uses of Computing

Half Term 6

- Project Development - Analysis
- Mock assessment preparation

Year 13:

Half Term 1

- Fundamentals of Communication and Networking
- Fundamentals of Algorithms
- Big Data
- Project Development - Design
- Project Development – Implementation

Half Term 2

- Fundamentals of Databases
- Project Development - Testing

Half Term 3

- Fundamentals of Functional Programming
- Pre-release Code: Exam Material

Half term 4

- Fundamentals of Computational thinking
- Pre-release Code: Exam Material
- Systematic Approach to problem solving.

Half Term 5

- Pre-release code: Exam Material
- Exam Preparation

Half Term 6

- External examination period