Computer Science: GCSE

Course Overview

- **Exam Board** – OCR
- **Usual Age Range** – 14-16
- **Qualification** –1 GCSE
- **Curriculum Time** – Five 50 minute lessons per week in class. This time will also include Careers Advice and Guidance sessions linked to Digital Technology roles as well as industry projects running throughout the year.
- **Assessment** – this curriculum is assessed via:
    - 2 x 90 minute exams
- **Grading** – 1 to 9
- **Full specification** - https://www.ocr.org.uk/qualifications/gcse/computer-science-j277-from-2020/specification-at-a-glance

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 useful knowledge, understanding and skills that can be applied in any Digital Technology setting in their future career and of particular use to students considering a career in computing, games development and cyber security.

The further intent of the Curriculum is to give students useful technical skills around programming such as the ability to write an effective flowchart, produce Pseudocode, develop an understanding for basic programming skills, understand the components that make up a digital system and develop their computational thinking.

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 addition as well as calculating file sizes for images and sound.

Suggested next step **destinations** after completion include A Level Computer Science, Level 3 Technical IT or Extended Project Qualification.

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 from additional study of Computer Science GCSE Theory for the exam revision and also if they are absent from the UTC but well enough to complete remote learning. It is also helpful to practise Python coding regularly.

- Seneca – https://bit.ly/2OSs4ZI
- CGP Study Guide – https://www.cgpbooks.co.uk/secondary-books/gcse/computer-science/cocub43-gcse-computer-science-ocr-revision
- Practice Assessments and papers - https://www.ocr.org.uk/qualifications/gcse/computer-science-j277-from-2020/assessment/
- Python Programming - https://www.codecademy.com/catalog/language/python

- Oak academy: https://classroom.thenational.academy/subjects-by-key-stage/key-stage-4/subjects/computing
- GCSE Bitesize: https://www.bbc.co.uk/bitesize/examspecs/zmtchbk
- Students can access all lesson materials on Teams along with a Python tutorial pack

Some other useful websites:

- Simple notes, diagrams and activities - https://www.bbc.co.uk/bitesize/examspecs/zmtchbk
- Video Guides mapped to topics - https://student.craigndave.org/J277
- Alternative web based revision sites - https://www.computerscience.gcse.guru/ and https://getrevising.co.uk/resources/level/gcse/subjects/computing
- Topic and module quizzes: https://quizizz.com/admin
- Revision notes - https://www.csnewbs.com/ocr-gcse
- Python course - https://time2code.today/python-course

Curriculum Overview

The learning in Computer Science (*1 GCSE*) is sequenced as follows.

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

Key Topics

- Systems architecture
- Memory and storage
- Computer networks, connections and protocols plus network security
- Systems software
- Ethical, legal, cultural and environmental impacts of digital technology
- Algorithms and Programming fundamentals
- Producing robust programs using Boolean logic
- Programming languages and Integrated Development Environments

Year 10:

Half Term 1

- Python coding: numbers and basic operations and functions
- Systems architectures,

Half Term 2

- Python coding: selection
- Memory and Storage including data representation: units of storage; binary and hexadecimal; characters, sound, images, compression

Half Term 3

- Python coding: data types and iteration
- Computer networks, connections and protocols

Half term 4

- Python coding: more iteration and handling user inputs
- Network security
- Systems software

Half Term 5

- Python coding: working with files and 1D and 2D arrays
- Ethical, legal, cultural and environmental impact of digital technology

Half Term 6

- Python challenges and problem solving
- Revision and mock exam preparation

Year 11:

Half Term 1

- Computational thinking
- Designing, creating and refining algorithms
- Searching and sorting algorithms
- Programming fundamentals

Half Term 2

- Data types
- Additional programming techniques

Half Term 3

- Defensive design
- Testing
- Boolean logic
- Languages
- The Integrated Development Environment

Half term 4

- Revision and exam preparation for Computer Systems 01

Half Term 5

- Revision and exam preparation for Computational thinking, algorithms and programming (02)

Half Term 6

- External examination period