

# Computer Science

## Curriculum Intent

Computer Science is becoming a popular subject at Turves Green Boys' School. Lessons are designed to enthuse pupils, prepare pupils for the future, build confidence and challenge all abilities.

Theoretical Computing – Throughout study pupils will develop their understanding of the technology used to power the modern day systems which we rely on. Developing understanding of networking, security, inner workings and the storage of data. In addition to this we provide extensive understanding surrounding the origins of computing and a breadth of history.

Programming – Pupils will develop their ability to recognise a problem, extract important information, design a solution to a problem, implement using code, test and evaluate fully. Pupils will be introduced to concepts including variables, sequence, arithmetic operators, Boolean operators, selection, iteration, file handling, regular expressions, time handling, random numbers, subroutines, and functions.

## KS3

Computer Science curriculum at TGBS covers lesson at Key Stage 3 for all pupils. It is our aim to equip pupils with the confidence and skills to be able to use knowledge that they will need at Key Stage 4 and in the wider world. Pupils study units that cover Digital Literacy, Information Technology and Computer Science these typically last for half a term. Assessment at Key stage 3 is mapped to key skills in Key Stage 4's Creative i-Media and Computer Science qualifications.

## Year 7

Using Computers safely, effectively, and responsibly

Create and navigate through folders knowing the difference between home and shared area. • Communicate via email. • Use attachments to add files onto an email. • Make use of tools to improve the accuracy of an email. • Relevant theme picked on a yearly basis to reflect areas of need regarding e-safety. • Understand the risks that exist online. • Know what to do if an issue occurs when on the internet. • Know how to prevent issues from happening when using the internet. • Describe the features that make a strong password.

Word and PowerPoint Skills

Style elements within a document to match setting. • Make use of headers/footers to better organise work. • Separate content using bullet points and numbering where appropriate. • Use keyboard shortcuts to optimise working. • Use images and tables to enhance content. • Make use of the slide master to make template slides with key elements copied across each slide.

## Programming in Scratch

Describe the difference between data types. • Use variables to store data. • Deciding what code to run based on conditions. • Repeating code to make program more efficient. • Creating programs based on a given problem.

## Hardware

Define the term peripheral, understanding the difference between input, output and storage peripherals. • Describe how some devices can be both input and output. • Identify the roles of different hardware devices that exist within a computer system. • Explain how certain devices can be used by less able users to assist with their day to-day lives. • Explain why binary numbers are used by computers • Convert between binary and denary representation of numbers • Add together binary numbers.

## Spreadsheets

Identify a cell's location via cell referencing. • Identify a group of cells using a range. • Format and structure data as appropriate in a spreadsheet. • Sort and search for data within a spreadsheet. • Use formulae to perform calculations on data. • Use formulae to more easily locate information from other worksheets. • Use conditional formatting to format certain elements based on a condition. • Model possible scenarios using spreadsheets.

## Year 8

### E-safety

E-safety • Relevant theme picked on a yearly basis to reflect areas of need regarding e-safety. • Understand the risks that exist online. • Know what to do if an issue occurs when on the internet. • Know how to prevent issues from happening when using the internet. • Describe the features that make a strong password.

### Cyber Security

Explain the differences between types of malware. • Describe threats other than malware that could compromise a computer system. • Explain the different

methods of protecting a computer system from malware and other threats. • Explain the role of encryption. • Use the Caesar Cipher to encrypt data. • Explain the flaws with the Caesar Cipher. • Scramble data using keyword encryption. • Explain the features that make a good keyword for encryption purposes.

## Data Representation

Recall how to convert between binary and denary numbers. • Recall how to add two binary numbers together. • Convert between hexadecimal, binary and denary numbers. • Recognise the similarities between how image and sound are stored within a computer. • Understand how compression affects a file, both in terms of quality and size. • Understand the link between resolution and file size in a computer.

## Python

Create programs to solve problems in a text-based language. • Describe the difference between data types. • Identify the correct data type to be used for a variable. • Use variables and inputs to take user entry and store it for later use in a program. • Use selection (if statements) to decide what code to run based on a condition. • Perform calculations using mathematical and comparative operators in a program. • Use iteration (for and while) to repeat sections of code. • Recognise errors when running code and correct them. • Solve given problems using Python to create programs.

## Website design

Create webpages based on a given brief. • Understand what makes a good house style. • Recognise the file types used on a webpage, both for the webpage itself and the content included. • Recall the impact on loading times that file size of content has. • Understand what makes up a domain name. • Link webpages together using hyperlinks and hotspots. • Use CSS to better format a webpage.

## Year 9

### Cyber Security

Explain the differences between types of malware. • Describe threats other than malware that could compromise a computer system. • Explain the different methods of protecting a computer system from malware and other threats. • Explain the role of encryption. • Use the Caesar Cipher to encrypt data. • Explain the flaws with the Caesar Cipher. • Scramble data using keyword encryption. • Explain the features that make a good keyword for encryption purposes.

## Data Representation

Recall how to convert between binary and denary numbers. • Recall how to add two binary numbers together. • Convert between hexadecimal, binary and denary numbers. • Recognise the similarities between how image and sound are stored within a computer. • Understand how compression affects a file, both in terms of quality and size. • Understand the link between resolution and file size in a computer.

## Python

Variables and constants • Data types • Decomposing a problem. • Abstracting information from a problem. • Creating flowcharts to model an algorithm. • Interpreting flowcharts • Create subroutines to be reused within a program. • Searching algorithms • Sorting algorithms

Create variables • Perform arithmetic operations • Use selection in Python • Create loops to repeat sections of code • Define the difference between count controlled and condition-controlled loops. • Create lists • Manipulating lists.

## Website design

Create webpages based on a given brief. • Understand what makes a good house style • Creating an index page for a website using Dreamweaver. • Link webpages together using hyperlinks and hotspots. • Testing the functionality of the website. • Evaluate the website.

## Multimedia presentation

Create a multimedia presentation based on a given brief. • Understand what makes a good house style. • Make use of the slide master to make template slides with key elements copied across each slide. • Incorporate multimedia elements such as audio, video, images, text, and animation. • Facilitate nonlinear functionality to the multimedia presentation. • Testing the functionality presentation. • Evaluate the multimedia presentation.

## **KS4**

The Computer Science curriculum is planned to fully cover exam specifications and prepare pupils for year 11. Assessment is used to make sure it is meaningful and efficient and links to GCSE from year 7-11. Key Stage 3 curriculum has been designed to build the pupils' interest in the subject, confidence, and curiosity in learning.

## Computer Science

J277/01: Computer systems

This component will assess:

1.1 Systems architecture

1.2 Memory and storage

1.3 Computer networks, connections and protocols

1.4 Network security

1.5 Systems software

1.6 Ethical, legal, cultural and environmental impacts of digital technology

J277/02: Computational thinking, algorithms and programming

This component will assess:

2.1 Algorithms

2.2 Programming fundamentals

2.3 Producing robust programs

2.4 Boolean logic

2.5 Programming languages and Integrated Development Environments

## **GCSE Exam**

Computer Science - OCR's GCSE (9-1) in Computer Science (J277)

Computer Science uses the GCSE 1-9 grading system, where 9 is the best grade. All examinations are terminal (at the end of Year 11, two compulsory components). The assessments are comprised of the following components:

Assessment Overview

Written paper: 1 hour and 30 minutes 50% of total GCSE

80 marks

This is a non-calculator paper. All questions are mandatory.

This paper consists of multiple choice questions, short response questions and extended response questions.

Written paper: 1 hour and 30 minutes 50% of total GCSE

80 marks

This is a non-calculator paper. This paper has two sections: Section A and Section B. Pupils must answer both sections.

All questions are mandatory. In Section B, questions assessing pupils' ability to write or refine algorithms must be answered using either the OCR Exam Reference Language or the high-level programming language they are familiar with.

Practical Programming skills

All pupils will be given the opportunity to undertake a programming task or tasks during their course of study. The programming task(s) will allow pupils to develop skills within the following areas when programming:

Design

Write

Test

Refine

Practical Programming skills will be assessed in Component 2 of the qualification, in particular Section B.

### **Career Prospects**

There are many different careers that Computer Science qualifications could help you move towards.

Here are some examples of sample career paths:

Programming and software Development, Project Management, Robotics, Artificial Intelligence, Web Developer, Game Designer, Network Administrator or Software Architect.