COMPUTER SCIENCE
NORTH OXFORDSHIRE ACADEMY

Bridging Activities: From GCSE to OCR A Level Computer Science

This document outlines bridging activities for students transitioning from the Pearson Edexcel GCSE (9–1) Computer Science to the OCR A Level Computer Science. Each section builds on GCSE content and introduces foundational A Level concepts to ensure a smooth progression.

Congratulations on picking OCR A Level computer science. I am really looking forward to teaching you next year.

You will see 7 different activities below. Please see which are optional and which are mandatory. Please complete the three mandatory activities as well as at least 2 of the optional. These will help build upon already taught knowledge from GCSE and help consolidate your learning ready for the A Level.

1. Strengthen Core Programming Skills (Mandatory)

Reinforce Python programming skills and introduce object-oriented programming and recursion.

- Suggested Activities:
    - Mini-projects using Python (e.g. quiz app, text-based game)
    - Debugging challenges with logic/syntax/runtime errors
    - Refactoring tasks to improve readability and efficiency
    - Intro to OOP: Create simple classes and objects
    - Recursion basics: Factorial, Fibonacci, simple search algorithms
- Example Activity:
    - Create a 'BankAccount' class in Python with methods to deposit, withdraw, and display balance. This introduces students to object-oriented programming concepts.

2. Deepen Understanding of Computational Thinking (Mandatory)

Develop algorithmic thinking and introduce formal analysis techniques like Big-O notation.

- Suggested Activities:
  - Trace table exercises with nested loops and conditionals
  - Algorithm comparison: Bubble vs Merge sort
  - Big-O introduction: Visualise time complexity with graphs
  - Pseudocode to code: Convert GCSE-style pseudocode into Python
- Example Activity:
  - Compare the number of steps required by Bubble Sort and Merge Sort for a list of 10 elements. Discuss the efficiency of each algorithm.

3. Expand Knowledge of Data Representation (Mandatory)

Build on binary and hexadecimal knowledge and explore image and sound representation.

- Suggested Activities:
  - Binary arithmetic: Addition, shifts, overflow
  - Hexadecimal puzzles: Decode messages or colour codes
  - Bitmap image analysis: Calculate file sizes
  - Sound sampling simulation: Create a basic WAV file from sample data
- Example Activity:
  - Calculate the file size of a 100x100 pixel image with 24-bit colour depth. This reinforces understanding of how images are stored digitally.

4. Explore Computer Architecture and Systems (Optional)

Introduce CPU components, instruction sets, and assembly language.

- Suggested Activities:
  - Fetch-decode-execute cycle animation or storyboard
  - Compare storage types: Create a decision tree
  - Intro to assembly: Use a simulator like Little Man Computer (LMC) https://www.101computing.net/lmc/

- Example Activity:
    - Use the LMC simulator to write a program that adds two numbers and outputs the result. This introduces students to low-level programming.

5. Investigate Networks and Cybersecurity (Optional)

Explore layered protocols, encryption, and network topologies.

- Suggested Activities:
    - Protocol roleplay: Simulate TCP/IP communication
    - Network design task: Plan a school network
    - Cyberattack case studies: Analyse real-world breaches
- Example Activity:
    - Design a secure school network using a star topology and explain how firewalls and encryption protect data.

6. Discuss Ethical, Legal, and Environmental Issues (Optional)

Critically evaluate the impact of computing on society.

- Suggested Activities:
    - Debates: 'Should AI be regulated?'
    - Research task: Investigate a tech company's environmental footprint
    - Case study analysis: GDPR violations or AI bias examples
- Example Activity:
    - Debate the ethical implications of facial recognition technology in public spaces. This encourages critical thinking and awareness of societal impact.

7. Bridging Projects (Optional)

Combine multiple GCSE topics into larger, open-ended projects.

- Suggested Activities:
    - Smart Home Simulation: Use Python to simulate sensors and automation
    - Cybersecurity Escape Room: Create puzzles based on encryption and malware
    - Data Visualisation: Use Python libraries to graph binary data or network traffic

- Example Activity:
    - Create a Python-based Smart Home system that simulates temperature sensors and automated lighting. This integrates programming, data handling, and system design.