Transition Subject Tasks

Subject: Computer Science

Course: A Level

Exam Board: OCR

Tasks below are to be completed before September 2020

1. Improve your Python skills by completing as many programming exercises using the following link: https://bit.ly/2Ujso3o

   You must bring with you, at least 10 completed challenges. There are hints on the link above.

2. Download the Computer Science specification and go through it: https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/

   In September, bring a list of areas for development based on the topics you need to improve.

3. Learn another programming language. You don't need to become an expert, but it would be useful for you to bring knowledge of HTML and CSS to the class in September: https://www.w3schools.com/

4. Regularly look at the news, social media and video platforms for up-to-date news on all things technology and computing. Use sources such as TechRadar, Hackster, Pybites, Geek, PCMag etc.

5. If you have the opportunity, watch the following documentaries about Computer Science:

   - ATARI: Game Over
   - We Are Legion: The Story of Hacktivists
   - Indie Game: The Movie

6. Look at example exam papers and give them a go yourself, mark schemes available here: https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/ so that you can get an understanding of what is required in the next two years.

7. Familiarise yourself and do some research on the following: (a) The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control: how this relates to assembly language programs. (b) The Fetch-Decode-Execute Cycle; including its effects on registers. (c) The factors affecting the

performance of the CPU: clock speed, number of cores, cache. (d) The use of pipelining in a processor to improve efficiency. (e) Von Neumann, Harvard and contemporary processor architecture. 8. Do some research on the 5 pillars of computational thinking. Thinking abstractly, ahead, procedurally, logically and concurrently. Understand these so that you can use it in your future Computer Science education. 9. Revisit Boolean Algebra, specifically: (a) Define problems using Boolean logic. See appendix 5d. (b) Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions. (c) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation. (d) Using logic gate diagrams and truth tables. See appendix 5d. (e) The logic associated with D type flip flops, half and full adders. 10. Start to research ideas for your programming project. This is a big undertaking; you need to come in with at least 3 ideas in September – this will allow us to look and make sure they meet requirements. The project needs to meet all the criteria listed on page 19 onwards in the specification. Your project must not be static, it must have fluidity. All data entered must not be your own. For example; a weather app takes data from various scientific APIs and shows it in their own way dependent on requirements.	