Independent Study

In Computer Science you should…

1. **After each lesson**
   After each lesson review your lesson notes and make a note of anything that you don't understand so that you can ask about this in the next lesson. Locate and read the relevant material in the OCR textbook (generally, one lesson is equivalent to one double page spread). Add anything extra or interesting to the notes in your exercise book and complete any outstanding questions or homework set in the lesson.

2. **Weekly**
   - **Review Lecture Notes:** Go through the notes taken during lectures or classes for both Computer Systems and Algorithmic Thinking papers. Ensure understanding of key concepts and clarify any doubts.
   - **Practice Coding:** Dedicate time to coding practice, focusing on algorithms, data structures, and problem-solving techniques relevant to both the NEA project and the Algorithmic Thinking paper.
   - **Read Supplementary Materials:** Read relevant articles, blogs, or textbooks to deepen understanding of complex topics or explore new developments in computer science. (Find a news site you like, or build a custom reddit profile dedicated to CS)
   - **Solve Past Papers:** Regularly attempt past examination papers, focusing on both computer systems and algorithmic thinking papers. Analyse answers to identify areas of improvement.
   - **NEA:** You should be working on your NEA project, both the coding element (spending 2 hours a week minimum) and the writeup (spending 2 hours a week minimum). Fill out and print out a weekly NEA diary explaining what progress you have made in the week on the NEA, and any barriers that you have had to overcome while coding.
   - You can be watching videos on areas that you need to revise from
     - https://student.craigndave.org/a-level-videos
   - You can be reading revision materials for upcoming or previous lessons from
     - https://www.physicsandmathstutor.com/computer-science-revision/a-level-ocr/
     - https://isaaccomputerscience.org/topics/a_level?examBoard=all&stage=all#ocr

3. **Fortnightly**
   - **Review NEA Progress:** Evaluate progress on the NEA coding project, identifying any challenges or areas needing improvement. Make necessary adjustments to project plans or code implementation.
   - **Create Concept Maps or Mind Maps:** Create visual representations of key concepts, algorithms, and system architectures to aid in understanding and retention.
   - **Practice Written Responses:** Practice writing detailed and concise responses to exam-style questions, focusing on clear explanations and effective use of technical terminology.

4. **Termly**
   - **Review NEA Progress:** Evaluate progress on the NEA coding project, identifying any challenges or areas needing improvement. Make necessary adjustments to project plans or code implementation.
   - **Participate in Discussion Forums:** Engage in online discussion forums or study groups to discuss challenging topics, share resources, and collaborate on problem-solving.
   - **Create Concept Maps or Mind Maps:** Create visual representations of key concepts, algorithms, and system architectures to aid in understanding and retention.
   - **Practice Written Responses:** Practice writing detailed and concise responses to exam-style questions, focusing on clear explanations and effective use of technical terminology.
   - **Coding practice:** Staying on top of your programming skills either through mini self-decided projects or working through the tutorials and challenges at:
     C#
     - https://www.w3schools.com/cs/index.php
     - https://www.w3resource.com/csharp-exercises/
     SQL

- https://www.w3schools.com/sql/default.asp
- https://www.w3resource.com/sql-exercises/

Python

- https://www.w3schools.com/python/default.asp
- https://www.w3resource.com/python-exercises/

5. Throughout the year

- **Review NEA Progress:** Evaluate progress on the NEA coding project, identifying any challenges or areas needing improvement. Make necessary adjustments to project plans or code implementation.
- **Participate in Discussion Forums:** Engage in online discussion forums or study groups to discuss challenging topics, share resources, and collaborate on problem-solving.
- **Create Concept Maps or Mind Maps:** Create visual representations of key concepts, algorithms, and system architectures to aid in understanding and retention.
- **Build Cyber Skills:** You can be working on your cybersecurity skills working through labs
  - https://www.immersivelabs.com/
- **Practice Written Responses:** Practice writing detailed and concise responses to exam-style questions, focusing on clear explanations and effective use of technical terminology.

Exam board specific papers: Complete past paper questions with mark schemes are made available on OCR
- https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/assessment/ You should work through these questions to test your knowledge and critically assess your work using the mark schemes. Aim for two every half term!

Other exam board papers: Access other exam board past paper questions and mark schemes to extend your knowledge further
AQA - https://www.aqa.org.uk/subjects/computer-science-and-it/as-and-a-level/computer-science-7516-7517/assessment-resources
Cambridge International - https://www.cambridgeinternational.org/programmes-and-qualifications/cambridge-international-as-and-a-level-computer-science-9618/past-papers/
WJEC - https://www.wjec.co.uk/qualifications/computer-science-as-a-level/#tab_pastpapers