Learning to fly in **Computer Science**

These are the skills and habits which outstanding students of Computer Science develop. Do as many of these as possible to become completely independent in the subject and to develop the skills and knowledge needed to attain 8/9 at GCSE and A* at A Level. **Remember that just doing what your teacher tells you - in your lessons and beyond - is not enough to develop your full potential in the subject.**

In your lessons

- Ask specific questions and ask for help.
- Think critically and computationally about your own work and find ways to improve it.
- Listen to other students and the teacher, but feel free to engage and share your viewpoint.
- Think about ways in which new knowledge you encounter in the lesson fits in with what you already know.
- Take notes to aid your memory.
- Be prepared to try new approaches to problem solving (Think laterally).
- Accept that programs will not work on your first attempt (90% of the time!)

Between your lessons

- Look back over your notes and activities from the last lesson.
- Review the work of a lesson, making sure you know what you have or haven't understood.
- Write down questions from your reading about concepts and ideas you don't understand to bring to your next lesson.
- Make notes of any questions to ask your teachers or look up.
- Go to see your teacher to ask for clarification and guidance.
- Practice programming concepts/techniques covered in class.

Beyond your lessons

- Try to practice programming. Programming is like learning a musical instrument, the more you practice the better you become. Exercises are on SharePoint or there are lots of free courses you can subscribe to. **E.g. Coursera, Code Academy, Hour of Code (Beyond an hour of code),**
- Stay informed about developments in digital technology by subscribing to online services like BBC Click, TechCrunch, and MIT. Additionally, watch high-quality science fiction films and tech-related documentaries on TV, and write a critical reaction to them.
- Keep a blog or vlog about your experiences with technology
- Play stimulating video games. Experience a variety of different game genres. Write a brief critical evaluation of a game, its purpose, goals, gameplay, graphics etc.
- Develop a sense of the big picture – technology is developing all the time, and you need to be aware of the latest developments.
- Subscribe to **MOOCs** (Massive Open Online courses). These are free and supplied by many Universities and exam boards. You can use these to supplement your notes. Such as **Harvard University Introduction to Computer Science**: https://www.edx.org/search?subject=Computer+Science
- Keep a vocabulary log to keep track of new words/terms you encounter or an app you use.

eXtension Zone

- **Alice** – 3D programming in Java: www.alice.org
- **BBC Bitesize** – Computer Science revision content: https://www.bbc.co.uk/bitesize/subjects/z34k7ty
- **BBC Make It Digital** – Articles and resources to help programmers flourish: www.bbc.co.uk/makeitdigital
- **Bebras** - Computational Challenge for students: https://bebras.uk/
- **British Informatics Olympiad** – Programming competition: www.olympiad.org.uk
- **Cambridge GCSE Computing Support** – Instructional videos for Computing students: www.cambridgegcsecomputing.org/new-course. Please register as a learner and create a free account to access the resources.
- **Cambridge University** – Pre-Computer Science Degree content https://www.cst.cam.ac.uk/freshers
- **Code.org** - Resources and information on learning to code: www.code.org
- **Codecademy** – Excellent site for learning various programming languages: www.codecademy.com
- **Codédex** - Gamified platform for learning coding and web development (Python, HTML5, JavaScript): https://www.codedex.io/home
- **Code Combat** – Coding games for various programming languages: https://codecombat.com/
- **GCHQ Turing Challenge** – Series of puzzles to enhance problem solving and logical thinking: https://www.gchq.gov.uk/information/turing-challenge
- **MIT Online Courses** – Open access MIT courses: https://ocw.mit.edu/search/
- **National Cipher Challenge** – University hosted codebreaking challenge: https://www.cipherchallenge.org/
- **The National Museum of Computing** – Bletchley Park: www.tnmoc.org
- **Perse Coding Challenge** – Coding challenge: https://pctc.perse.co.uk/
- **Raspberry Pi** – News and resources around the Raspberry Pi: https://www.raspberrypi.org/
    - **Adafruit Learning System** – Raspberry Pi projects and resources: http://learn.adafruit.com/category/raspberry-pi
    - **Coder** – Raspberry Pi web coding project (HTML5, JavaScript, CSS): http://googlecreativelab.github.io/coder/
    - **MagPi** – Projects and resources for the Raspberry Pi: https://magpi.raspberrypi.com/
- **Web Development** – Online resource to support web development studies. www.w3schools.com

Of course, we recognise that our students have busy lives and that this level of engagement is not always possible all the time – but this is what you should aim for if you want to reach the highest level in the subject. You don’t need to do ALL these things to improve – just one or two of them will have an impact. Decide on two or three things to focus on to improve your skills.