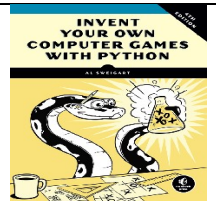
Writing Code

There are many good resources for learning to write code and to practice your coding skills.

Here are some you might like to try. Your teacher may recommend particular resources, or direct you to complete specific exercises.

	Think Python Learning with Python 3 Peter Wentworth, Jeffrey Elkner, Allen B. Downey, Chris Meyers Oct. 2012 http://openbookproject.net/thinkcs/python/english3e/index.html	One of the best free books on learning to program using python. The emphasis is on understanding why we write code and solve problems in a particular way, which is useful for A-level students. The book is well organised, with plenty of exercises in each chapter, plus a glossary of key words. Up to Ch14 is AS level, the rest of the book covers A level standard code, including the key data structures and algorithms. Note that the same resource is available for other languages, namely *Think Java* and *Think C*
	Invent with Python Albert Sweigart http://inventwithpython.com/	A nice way to start python, this site has a collection of introductory books on writing code, also all free! Each chapter has a game (or similar to make) and includes the full code, plus a step-by-step walkthrough of how to make it. It is a good exercise to read code before you write it, so making some of these games is useful.
British Informatics Olympiad	The British Informatics Olympiad https://www.olympiad.org.uk/problems.html	Lots of hard coding challenges. Like the maths challenge, only for programming! The Mayan Calendar is a good starting point.

There are also various PiXL resources to teach basic python up to GCSE standard.

The coding challenges below will let you check your skills. Part of the transition to A-level is combining skills, and also ensuring that you plan and test your work thoroughly, so think about how you can re-use components and design your code for readability and robustness.

1. Write an program to:
    a. Ask the user to input
        i. Their first name
        ii. Their surname
        iii. A date, in the format DD/MM/YYYY
    b. The program should then output a customer ID as follows:
        i. The date in the format YYYYMMDD, then the first three letters of the surname, then the first initial, then the length of their first name. All letters should be in capitals
        ii. For example, John Smith, 27/05/2017 would give 20170527SMITHJ4
    c. The program should validate any inputs and keep asking for inputs until the user enters correct details or types "quit" at any point

Plan your algorithm first, using a flowchart or pseudocode

Code your algorithm, and provide evidence of both your code and the working output

Create a test plan for your algorithm, including testing your validation with normal, boundary and erroneous data

2. Write a program to:
    a. Ask the user to input
        i. The name of a product
        ii. Its cost in pounds
        iii. The program should keep asking for inputs until the user types "None"
    b. The program should then output:
        i. The name and price of the most expensive item
        ii. The name and price of the least expensive item
        iii. The average price of the items
        iv. The total cost of the items
            1. Items over £50 get a 5% discount
            2. VAT is added at the end at 20%
    c. The program should validate any inputs

Plan your algorithm first, using a flowchart or pseudocode

Code your algorithm, and provide evidence of both your code and the working output

Create a test plan for your algorithm, including testing your validation with normal, boundary and erroneous data.