****

**A-Level Computer Science Guide**

**How Computer Science will be taught:**

A level Computer Science is taught via face-to-face teaching of the narrative; this is by either presentation or lecture. There will also be a large number of practical lessons where you will be required to develop suitable solutions for a given problem. The practical programming lessons will develop your skills for both the NEA and the paper 1 exam. At the start of the course you will complete an online assessment of previous knowledge

**Working expectations:**

You are expected to put at least as many hours into A level Computer Science outside of lessons as you would in lessons. A level Computer Science requires you to develop skills in different programming languages and challenge yourselves to create different programs. You will be required to read around the subject and look at current issues associated with Computer Science. You will need to submit work onto the moodle and complete all homework to the required deadlines. In year 12 you will begin your NEA (coursework project); this has a set of deadlines that must be met. The NEA is an independent project where you choose to either solve a problem or carry out an investigation.

**What 100% effort in this subject looks like:**

* As much time out of lesson devoted to Computer Science as you are assigned in lessons.
* Using the moodle to look ahead to the next lesson and reading ahead in the booklets to prepare.
* Working through Exam Pro questions, requesting topics as needed.
* Making your own revision booklets.
* Using pre-release materials to prepare for paper 1 examination.
* Focus on the NEA during the summer to meet the deadlines set.

**Folder Policy:**

*Your folder should have:*

1. Practice code in Python
2. Pre release materials – code solutions
3. Structured notes for each of the topics
4. Exam practice papers
5. NEA report and program
6. Past test paper

The majority of the work will be saved on the Moodle, however any written responses or drafts of your NEA needs to be stored in an assessment folder which will be checked periodically.

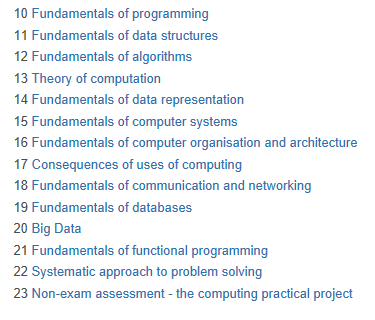
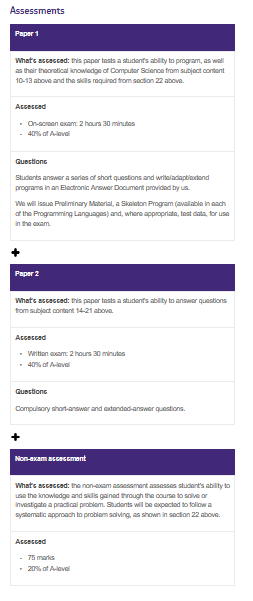
**What Marking looks like:**

* Ordinary class notes are not marked, as there is nothing to assess
* Homework’s that are based on past exam questions will be assessed and graded
* NEA will be assessed at key deadline points and marked against grading criteria
* Tasks and exam prep will be marked with feedback for using technical terms

**What Homework looks like:**

* Coding a solution
* Research tasks
* Exam questions
* Reading and research

**Specification at a glance:**



**Summer preparation**

**GCSE Computer Science**

**Watch:**

**Craig and Dave GCSE Videos**

<https://www.youtube.com/channel/UC0HzEBLlJxlrwBAHJ5S9JQg/search?query=aqa+gcse>

In preparation for the A level, watch the series of videos made for the GCSE course, make brief notes and summarise each video

Do:-

Create a report investigating Computer Science. In the report include aspects of computational thinking, Look at the different skills including decomposition, abstraction and logical thinking skills

Alan Turing has been accredited with the founder of modern day Computer Science. Research Alan Turing and describe how his work has been the foundation for computers today

**Programming**

To develop and continue programming skills

Katarina is developing a two-player dice game.

The players roll two 6-sided dice each and get points depending on what they

roll. There are 5 rounds in a game. In each round, each player rolls the two dice.

The rules are:

• The points rolled on each player’s dice are added to their score.

• If the total is an even number, an additional 10 points are added to their score.

• If the total is an odd number, 5 points are subtracted from their score.

• If they roll a double, they get to roll one extra die and get the number of points rolled added to their score.

• The score of a player cannot go below 0 at any point.

• The person with the highest score at the end of the 5 rounds wins.

• If both players have the same score at the end of the 5 rounds, they each roll 1 die and

whoever gets the highest score wins (this repeats until someone wins).

Only authorised players are allowed to play the game. Where appropriate, input from the user should be validated.

Design, develop, test and evaluate a program that:

1. Allows two players to enter their details, which are then authenticated to ensure that they are

authorised players.

2. Allows each player to roll two 6-sided dice.

3. Calculates and outputs the points for each round and each player’s total score.

4. Allows the players to play 5 rounds.

5. If both players have the same score after 5 rounds, allows each player to roll 1 die each until someone wins.

6. Outputs who has won at the end of the 5 rounds.

7. Stores the winner’s score, and their name, in an external file.

8. Displays the score and player name of the top 5 winning scores from the external file