ICT & Computing Department Information for Yr 12

AQA A-Level Computer Science

7516 (AS-Level)
7517 (A-Level)

2016—2017

Aims and Objectives

The AQA A-Level Computer Science qualification offers learners the opportunity to follow a programme of study to:

- Develop programming skills.
- Develop an understanding of the system life cycle.
- Develop problem solving skills appropriate for work across multiple industries.
- Achieve a nationally recognised academic qualification.

Learners will also have the opportunity to acquire the essential knowledge and tools for the world of work by developing transferable skills such as planning, research and analysis, working with others and effective communication.

UCAS tariff points

The table below shows the UCAS tariff points awarded for the A-Level Computer Science qualifications.

Qualification	GLH	UCAS Points
AQA AS-Level Computer Science (7516)	360	A = 20 points B = 16 points C = 12 points D = 10 points E = 6 points
AQA A-Level Computer Science (7517)	720	A* = 56 points A = 48 points B = 40 points C = 32 points D = 24 points E = 16 points

Exam grades from the AS-Level do not contribute towards the A-Level, however the course has been designed so that all AS content will be assessed in the A-Level paper.

AS Topics

Unit	Topics	Assessment	Value
Paper 1	1. Fundamentals of programming 2. Fundamentals of data structures 3. Systematic approach to problem solving 4. Theory of computation	On Screen Exam 1 hour 30 minutes	50% of AS
Paper 2	5. Fundamentals of data representation 6. Fundamentals of computer systems 7. Fundamentals of computer organisation and architecture 8. Consequences of uses of computing 9. Fundamentals of communication and networking.	Written Exam 1 hour 30 minutes	50% of AS

A-Level (Year 13) Topics

Unit	Topics	Assessment	Value
Paper 1	10. Fundamentals of programming 11. Fundamentals of data structures 12. Systematic approach to problem solving 13. Theory of computation	On Screen Exam 2 hours 30 minutes	40% of A-Level
Paper 2	14. Fundamentals of data representation 15. Fundamentals of computer systems 16. Fundamentals of computer organisation and architecture 17. Consequences of uses of computing 18. Fundamentals of communication and networking. 19. Fundamentals of databases 20. Big Data 21. Fundamentals of functional programming 22. Systematic approach to problem solving	Written Exam 2 hour 30 minutes	40% of A-Level
Non-Exam Assessment	23. Students create a programme about an area of interest to them.	Coursework (Externally Moderated)	20% of A-Level

Deadlines

Unit	Teacher	Start Date	Deadline/Exam Date
AS Paper 1	Mr Robinson	Monday 12th September 2016	Monday 5th June 2017 9am
AS Paper 2	Mr Robinson	Monday 12th September 2016	Friday 9th June 2017 9am
A-Level Paper 1		September 2017	June 2018
A-Level Paper 2		September 2017	June 2018
A-Level Non-Exam Assessment		June 2017	December 2017

The preparatory work

Transitional tasks

Paper 1

Option 1 - Python task

If you have completed GCSE Computing using Python, complete the task below from the 2010 past paper. You will need to install Python 3 if you have not already installed it. https://www.python.org/downloads/release/python-352/

Question 6

Create a folder/directory **Question6** for your new program.

The variable table, **Table 2**, and the Structured English algorithm, **Figure 4**, describe a simplified version of a noughts and crosses match. A match consists of a user-specified number of games. In this simplified version, the two players complete each game on paper and then enter information about the result of each game into a program that totals the number of games won by each player. Assume that all games have a winner – there are no drawn games.

Table 2

Identifier	Data Type	Purpose
NoOfGamesInMatch	Integer	Stores the number of games in the match (specified by user)
NoOfGamesPlayed	Integer	Stores the number of games played so far
PlayerOneScore	Integer	Stores the number of games won by Player One
PlayerTwoScore	Integer	Stores the number of games won by Player Two
PlayerOneWinsGame	Char	Stores a 'Y' if Player One won the game and 'N' otherwise

Figure 4

```
PlayerOneScore ← 0
PlayerTwoScore ← 0
OUTPUT "How many games?"
INPUT NoOfGamesInMatch
FOR NoOfGamesPlayed ← 1 TO NoOfGamesInMatch Do
  OUTPUT "Did Player One win the game (enter Y or N)?"
  INPUT PlayerOneWinsGame
  IF PlayerOneWinsGame = 'Y'
    THEN PlayerOneScore ← PlayerOneScore + 1
    ELSE PlayerTwoScore ← PlayerTwoScore + 1
  ENDIF
ENDFOR
OUTPUT PlayerOneScore
OUTPUT PlayerTwoScore
```

What you need to do

Write a program for the above algorithm.

Test the program by showing the results of a match consisting of three games where Player One wins the first game and Player Two wins the second and third games.

Save the program in your new **Question6** folder/directory.

Evidence that you need to provide
Include the following in your Electronic Answer Document.

1 6	Your PROGRAM SOURCE CODE.	*(9 marks)*
1 7	SCREEN CAPTURE(S) for the test described above.	*(4 marks)*

Option 2

If you have not previously used Python, complete Tasks 1 to 3 on Codecademy
https://www.codecademy.com/learn/python

Paper 2

Task 2
Write a report on Data Encryption

Research the Caesar Cipher and Vernam Cipher.

For each Cipher, discuss:

- The origins of the cipher. Who created it and when.
- The algorithms used to encrypt & decrypt.
- An evaluation of the effectiveness.
    - How easy is it to 'crack'?

A bibliography must be included.

If you have any questions or concerns that you would like to discuss, please contact:

Stephen Robinson (Head of ICT & Computer Science)

Tel: 01480 459581
Email: srobinson@stpeters.cambs.sch.uk