
A Level Computer Science – Summer assignment

Note: The summer assignment must be handed in, **week beginning 3rd September**

Which exam board do we do?

The exam board is **AQA** and you can find information about the course on their site at:

<http://www.aqa.org.uk/subjects/computer-science-and-it/as-and-a-level/computer-science-7516-7517>

You will find past papers and the full specification on this website. The specification can be found at: <http://filestore.aqa.org.uk/resources/computing/specifications/AQA-7516-7517-SP-2015.PDF>

How much time do I need to spend on independent study?

Your teacher will set homework/independent study tasks each week and will guide you as to how much time to spend. 4 hours in total is the usual expectation. Here is an example of a typical weekly schedule

1. Writing tasks **2hrs**

2. Programming **2hrs** (although you will find that more time to improve programming skills is needed)

AQA Computer Science A Level assessment

Subject content

- 10 [Fundamentals of programming](#) (page 37)
- 11 [Fundamentals of data structures](#) (page 43)
- 12 [Fundamentals of algorithms](#) (page 48)
- 13 [Theory of computation](#) (page 50)
- 14 [Fundamentals of data representation](#) (page 58)
- 15 [Fundamentals of computer systems](#) (page 67)
- 16 [Fundamentals of computer organisation and architecture](#) (page 70)
- 17 [Consequences of uses of computing](#) (page 75)
- 18 [Fundamentals of communication and networking](#) (page 76)
- 19 [Fundamentals of databases](#) (page 82)
- 20 [Big Data](#) (page 83)
- 21 [Fundamentals of functional programming](#) (page 84)
- 22 [Systematic approach to problem solving](#) (page 87)
- 23 [Non-exam assessment – the computing practical project](#) (page 89)

Assessments

| Paper 1 | + | Paper 2 | + | Non-exam assessment |
|---|---|---|---|---|
| What's assessed This paper tests a student's ability to program, as well as their theoretical knowledge of computer science from subject content 10–13 above and the skills required from section 22 above. | | What's assessed This paper tests a student's ability to answer questions from subject content 14–21 above. | | What's assessed The non-exam assessment assesses student's ability to use the knowledge and skills gained through the course to solve or investigate a practical problem. Students will be expected to follow a systematic approach to problem solving, as shown in section 22 above. |
| Assessed <ul style="list-style-type: none">• On-screen exam: 2 hours 30 minutes• 40% of A-level | | Assessed <ul style="list-style-type: none">• Written exam: 2 hours 30 minutes• 40% of A-level | | Assessed <ul style="list-style-type: none">• 75 marks• 20% of A-level |
| Questions Students answer a series of short questions and write/adapt/extend programs in an electronic answer document provided by us. We will issue preliminary material, a skeleton program (available in each of the programming languages) and, where appropriate, test data, for use in the exam. | | Questions Compulsory short-answer and extended-answer questions. | | |

Using an appropriate programming language of your choice, you will plan and carry out a programming solution to solve the following problem:

Names for numbers

Sara Johnson teaches at Parkview Vale Primary School. She is concerned that her pupils are having difficulty in writing down numbers when they are said out loud. For example, when Sara asked the pupils to write down one thousand, nine hundred and five, many of them wrote down incorrect numbers.

Sara thinks that the pupils would benefit from more practice converting the names of numbers into figures. She would like a system that would produce a number written as a word, and require the pupil to enter the correct digits.

The system should present each pupil with ten numbers written as words and should keep a record of the correct responses. Sara would like the system to output the pupil's results each time a test is taken and produce an average (mean) mark each time the pupil has completed three tests.

Sara has also decided that once a pupil has answered a question successfully, that question should not appear in the ten questions the next time the pupil uses the system. If the question was answered incorrectly the question should appear the next time the pupil uses the system.

Sara has asked you to create a computer-based system that will:

- allow each pupil to be presented with 10 relevant numbers written as words per session
- ensure that questions that have been answered correctly by that pupil in previous sessions are not included
- store the pupil's responses to the questions
- calculate and display the number of correct responses
- display the average (mean) score each time a pupil has completed three sessions.

Miss Johnson's names for numbers sample questions.

| Name | Number |
|---|--------|
| One thousand, nine hundred and sixty five | 1,965 |
| Five thousand, seven hundred and seventy one | 5,771 |
| Nine thousand, eight hundred and sixty four | 9,864 |
| Eight thousand, eight hundred and sixty eight | 8,868 |
| Four thousand, six hundred and fifty four | 4,654 |
| Nine thousand, one hundred and thirty two | 9,132 |
| Seven thousand, two hundred and twenty seven | 7,227 |
| Six thousand, four hundred and nine | 6,409 |
| Four thousand and ninety nine | 4,099 |
| Two hundred and twenty eight | 228 |

Help if you use Python

There are a number of window toolkits available for Python. Tkinter comes installed as part of Python. It's quite straight forward to use but doesn't have as many different user interface components as others (e.g. a table view widget).

wxPython and PyQt are both cross platform window toolkits that can be used with Python but are only available as a separate download. Both of these come with GUI designers which allow you to drag and drop widgets onto a form so you can build the user interface without writing any code. You do, however, have to figure out how to import these into a Python project and make them do something. This part is more complex than sticking with the default Tkinter libraries but challenge is a good thing!

The final option is the Jython which allows you to write programs that use Java window components entirely in Python. It is also a separate download. Of course, you may want to go straight to using Java as learning a new language is always a good challenge and it's not nearly as difficult as learning your first!

Help if you use Java

If you are more interested in getting your feet wet with GUI design then Java is a good choice. Whilst the syntax for Java can seem a bit daunting at first, it is actually quite easy to create a simple user interface. Here is the download page for Java Development Kit:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

You can also choose whether to use a plain text editor or a beginner IDE for Java such as:

<http://www.bluej.org/>

Whilst it's probably not the best tutorial in the world, it is the official

one: <http://docs.oracle.com/javase/tutorial/getStarted/index.html>

Here is a very minimal Java GUI program (just 19 lines). It actually does quite a lot; it creates a GUI window, adds a button, registers the button to listen for user clicks, displays the window and prints a message when the button is pressed. Quite impressive for so few lines of code!

If you need guidance you can email: jblack@fromecollege.org

```
import javax.swing.*;
import java.awt.event.*;
public class MyWindow extends JFrame implements ActionListener{
    public MyWindow(){
        this.setSize(200, 100);

        JButton b = new JButton("Hello");
        this.getContentPane().add(b);
        b.addActionListener(this);

        this.setVisible(true);
    }
    public void actionPerformed(ActionEvent e){
        System.out.println("You pressed the button");
    }
    public static void main(String[] args){
        new MyWindow();
    }
}
```