

# GET READY TO STUDY A LEVEL COMPUTER SCIENCE

If you are planning to study A Level Computer Science with us in September, please review this document and complete the required activities. Please bring the completed activities with you at induction.

## FAQ

### **What specification will I study?**

AQA A-level Computer Science ([Specification at a glance](#))

### **Who can I contact if I have a question about this subject?**

Alex Melhuish [a.melhuish@barnsley.ac.uk](mailto:a.melhuish@barnsley.ac.uk)

### **What subjects go well with Computer Science?**

Maths, Further Maths, Physics

### **What grades should I have?**

In addition to the general sixth form entry requirements, learners must have GCSE grade 6 in Mathematics. You should also be planning to study Mathematics at A level.

# WHAT WILL I STUDY?

## **In Year 1, you will study the following topics:**

- Procedural Programming & Object Orientated Programming (Paper 1)
- Data Structures (Paper 1)
- Binary Mathematics (Paper 2)
- Data Representation (Paper 2)
- Hardware & Software (Paper 2)
- Communication and Networks (Paper 2)
- Logic Gates & Boolean Algebra (Paper 2)
- Input & Output Devices (Paper 2)
- Non-exam coursework (NEA)

## **In Year 2, you will study the following topics:**

- Databases (Paper 2)
- Algorithms (Paper 1)
- Recursion (Paper 1)
- Consequences of Computing (Paper 2)
- Computer Architecture (Paper 2)
- Data Structures (Paper 1)
- Regular Languages (Paper 1)
- Big O Notation (Paper 1)
- Functional Programming (Paper 2)
- Big Data (Paper 2)
- Floating Point Representation (Paper 2)
- Non-exam coursework (NEA)

# HOW WILL I BE ASSESSED?

## Paper 1

**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.

### Assessed

- On-screen exam: 2 hours 30 minutes
- 40% 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.



## Paper 2

**What's assessed:** this paper tests a student's ability to answer questions from subject content 14-21 above.

### Assessed

- Written exam: 2 hours 30 minutes
- 40% of A-level

### Questions

Compulsory short-answer and extended-answer questions.



## Non-exam assessment

**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

- 75 marks
- 20% of A-level

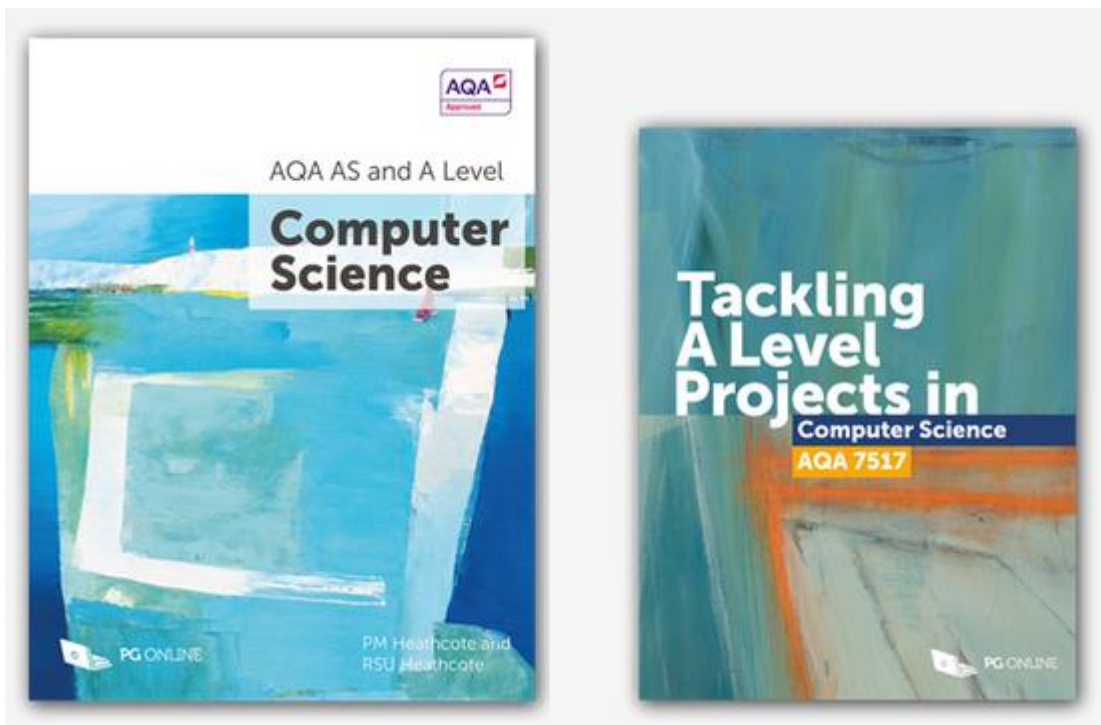
# WHAT WILL I NEED?

To study the course you will need the following equipment:

- A personal computer (desktop preferred)
- Windows Operating System installed
- Visual Studio Community Edition 2022 installed  
<https://visualstudio.microsoft.com/vs/community/>
- Internet connection
- Mini-whiteboard and pen
- A calculator
- A folder with dividers
- Pens and pencils
- Highlighters
- A ruler

We recommend all students buy the following textbooks.

<https://www.pgonline.co.uk/resources/computer-science/a-level-aqa/?tab=textbooks>



# FIND OUT MORE

These activities are to help broaden your understanding of the subject in preparation for studying this subject at an advanced level.

Careers	<ul style="list-style-type: none"><li>• <a href="#">Application analyst</a></li><li>• <a href="#">Applications developer</a></li><li>• <a href="#">Cyber security analyst</a></li><li>• <a href="#">Data analyst</a></li><li>• <a href="#">Forensic computer analyst</a></li><li>• <a href="#">Game designer</a></li><li>• <a href="#">Games developer</a></li><li>• <a href="#">Machine learning engineer</a></li><li>• <a href="#">Penetration tester</a></li><li>• <a href="#">Software engineer</a></li><li>• <a href="#">Systems analyst</a></li><li>• <a href="#">UX designer</a></li><li>• <a href="#">Web designer</a></li><li>• <a href="#">Web developer</a></li></ul>
YouTube	<p>Craig N Dave <a href="https://YouTube/CnDAQAComputerScience">https://YouTube/CnDAQAComputerScience</a></p> <p>Computerphile <a href="https://www.youtube.com/user/Computerphile">https://www.youtube.com/user/Computerphile</a></p>
Further Reading / Useful websites	<p>Isaac Computer Science <a href="https://isaacomputerscience.org/topics">https://isaacomputerscience.org/topics</a></p> <p>Maths &amp; Physics Tutor <a href="https://www.physicsandmathstutor.com/computer-science-revision/a-level-aqa/">https://www.physicsandmathstutor.com/computer-science-revision/a-level-aqa/</a></p>

# REQUIRED ACTIVITIES

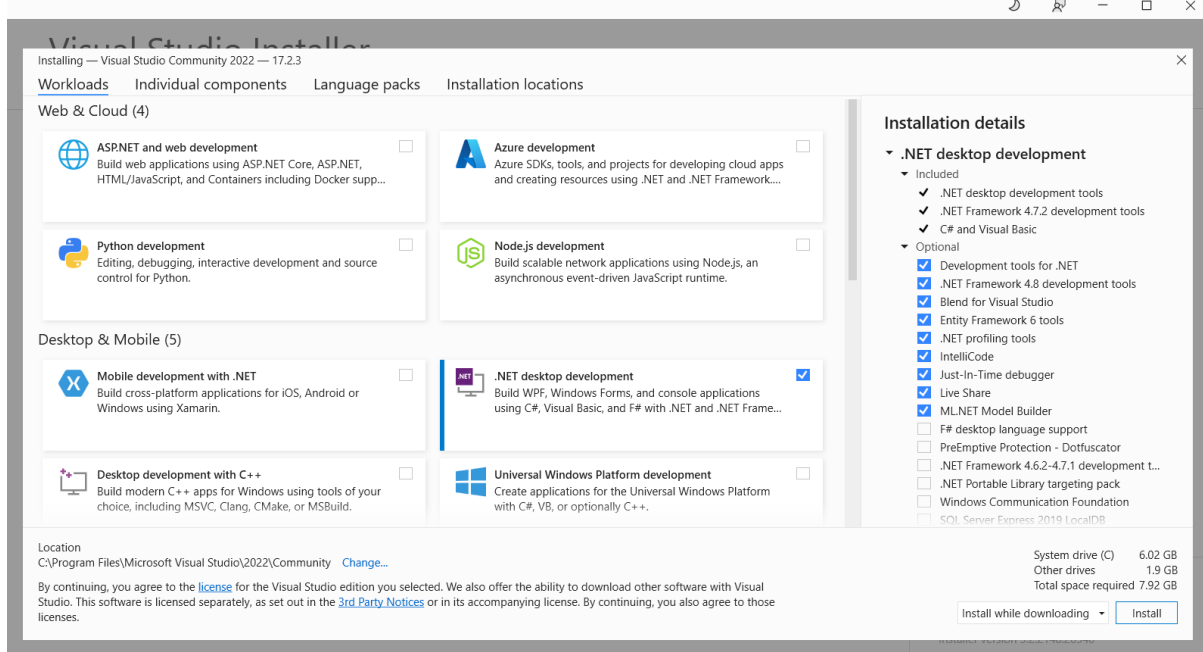
It is important that all the required activities are completed in preparation for starting your course. Please bring these completed activities with you at induction.

Choosing your A Levels can be a challenge for some learners therefore if you are undecided around which subjects you are planning to study completing these activities will give yourself greater insight into the course to help ensure you have made the right choice.

# INDUCTION TASK

## 1) Download and install Visual Studio 2022 (Community Edition)

- 1) Download and install Visual Studio  
<https://visualstudio.microsoft.com/vs/community/>
- 2) Select .NET desktop development when prompted



## 2) Create a project

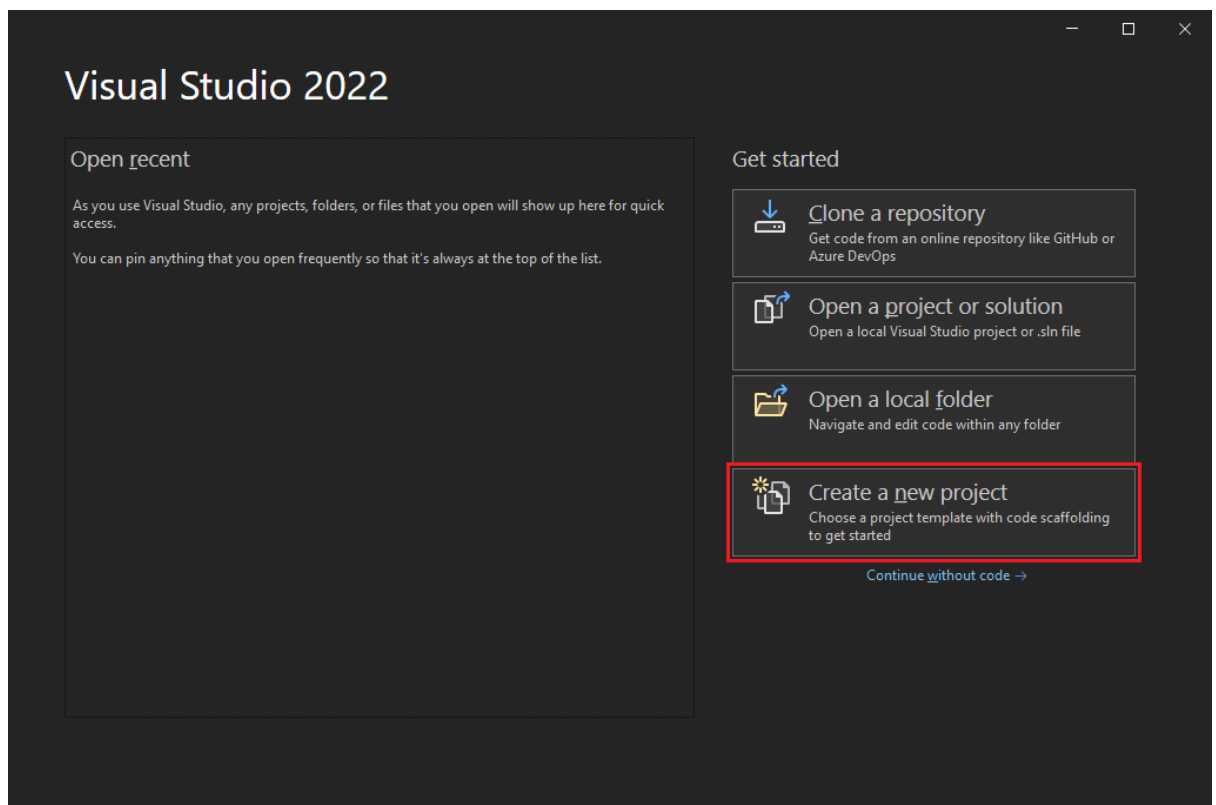
In this tutorial, you use Visual Studio to create and run a C# console app, and explore some features of the Visual Studio integrated development environment (IDE). This tutorial is part 1 of a two-part tutorial series.

In this tutorial, you:

- Create a Visual Studio project.
- Create a C# console app.

To start, create a C# application project. The project type comes with all the template files you need.

1. Open Visual Studio, and choose **Create a new project** in the Start window.



2. In the **Create a new project** window, select **All languages**, and then choose **C#** from the dropdown list. Choose **Windows** from the **All platforms** list, and choose **Console** from the **All project types** list.

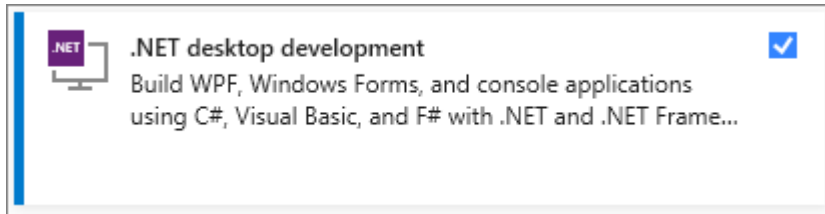
After you apply the language, platform, and project type filters, choose the **Console App** template, and then select **Next**.

Note

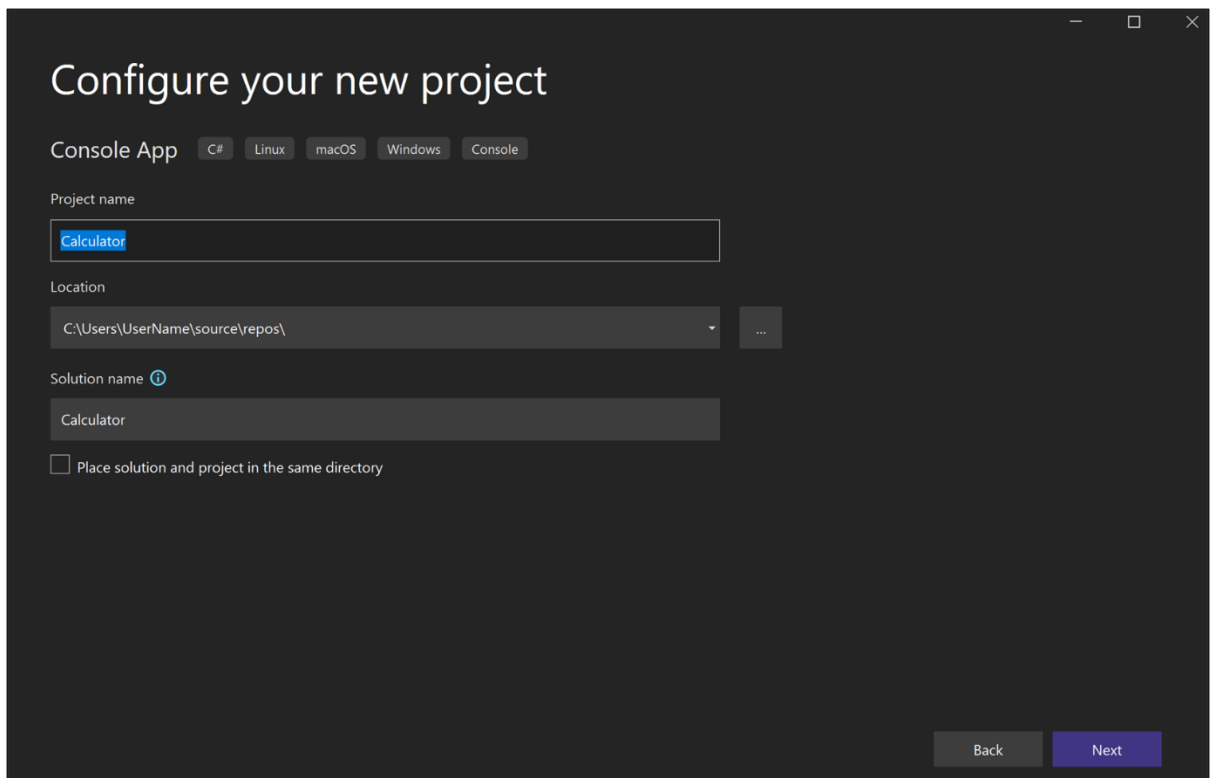
If you don't see the **Console App** template, select **Install more tools and features**.

Not finding what you're looking for?  
Install more tools and features

In the Visual Studio Installer, choose the **.NET desktop development** workload, and then select **Modify**. **(If you have not already done this!)**

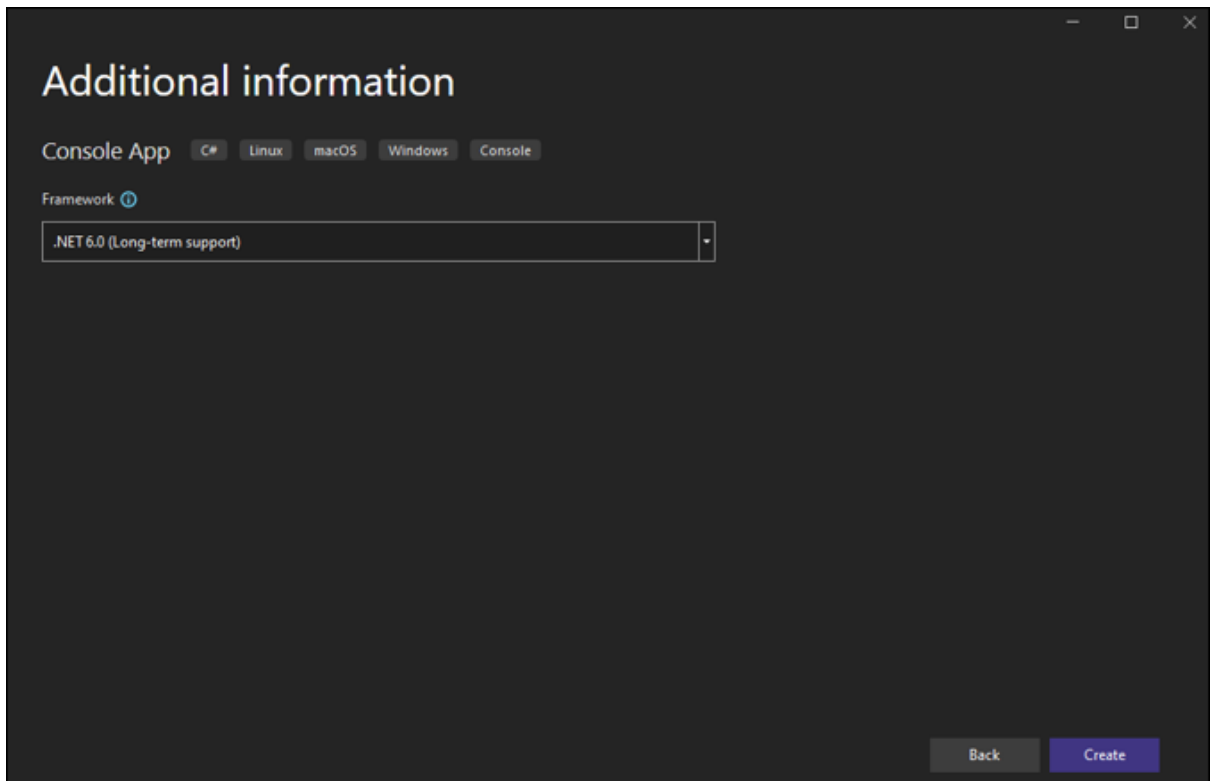


3. In the **Configure your new project** window, type or enter *Calculator* in the **Project name** box, and then select **Next**.



4. In the **Additional information** window, **.NET 6.0** should already be selected for your target framework. Select **Create**.





Visual Studio opens your new project, which includes default "Hello World" code.

To view it in the editor, select the code file *Program.cs* in the Solution Explorer window, which is typically on the right-hand side of Visual Studio.

The single code statement calls the [WriteLine](#) method to display the literal string "Hello, World!" in the console window. If you press F5, you can run the default program in Debug mode. After the application runs in the debugger, the console window stays open. Press any key to close the console window.

**\*Take a photo of your workstation monitor with Visual Studio in view add it to your evidence log (end of booklet)\***

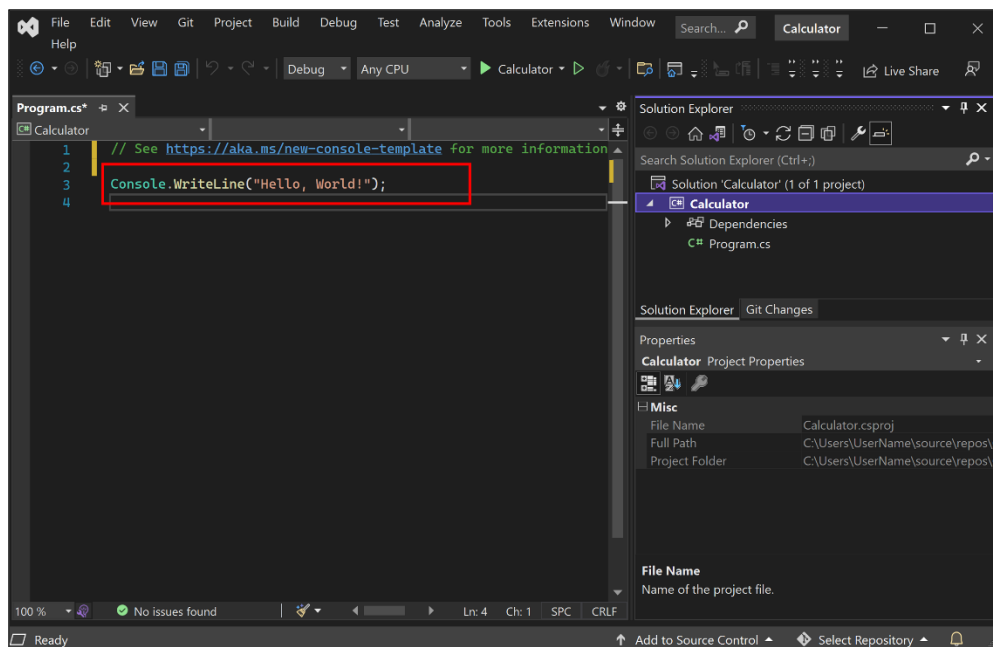
### 3) Create the app

In this section, you:

- Explore some basic integer math in C#.
- Add code to create a basic calculator app.
- Debug the app to find and fix errors.
- Refine the code to make it more efficient.

Start with some basic integer math in C#.

1. In **Solution Explorer**, in the right pane, select **Program.cs** to display the file in the code editor
2. In the code editor, replace the default "Hello World" code that says `Console.WriteLine("Hello World!");`.



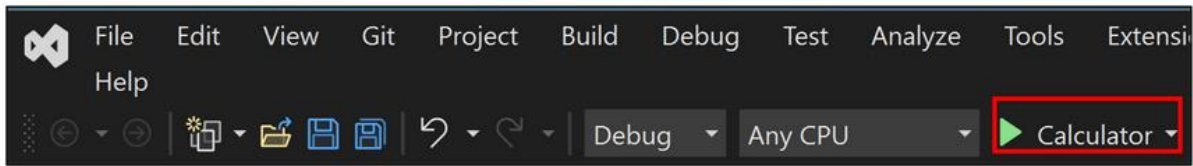
Replace the line with the following code:

Replace the line with the following code:

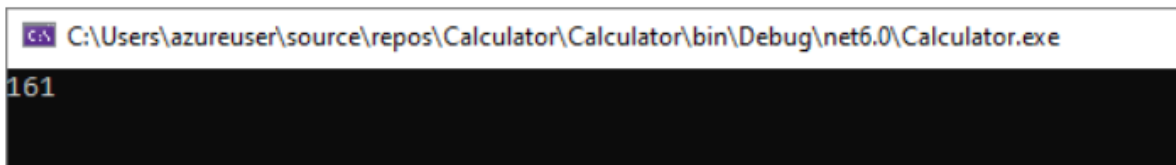
```
C# Copy  
  
int a = 42;  
int b = 119;  
int c = a + b;  
Console.WriteLine(c);  
Console.ReadKey();
```

If you type the code, the Visual Studio IntelliSense feature offers you the option to autocomplete the entry.

3. To build and run your app, press **F5**, or select the green arrow next to the name **Calculator** in the top toolbar.



A console window opens that shows the sum of  $42 + 119$ , which is **161**.



**\*Add this screenshot to your evidence log\***

4. Close the console window.
5. Optionally, you can change the operator to change the result. For example, you can change the `+` operator in the `int c = a + b;` line of code to `-` for subtraction, `*` for multiplication, or `/` for division. When you run the app, the result changes accordingly.

## 4) Create a calculator application

```
C#

// Declare variables and then initialize to zero.
int num1 = 0; int num2 = 0;

// Display title as the C# console calculator app.
Console.WriteLine("Console Calculator in C#\r");
Console.WriteLine("-----\n");

// Ask the user to type the first number.
Console.WriteLine("Type a number, and then press Enter");
num1 = Convert.ToInt32(Console.ReadLine());

// Ask the user to type the second number.
Console.WriteLine("Type another number, and then press Enter");
num2 = Convert.ToInt32(Console.ReadLine());

// Ask the user to choose an option.
Console.WriteLine("Choose an option from the following list:");
Console.WriteLine("\ta - Add");
Console.WriteLine("\ts - Subtract");
Console.WriteLine("\tm - Multiply");
Console.WriteLine("\td - Divide");
Console.Write("Your option? ");

// Use a switch statement to do the math.
switch (Console.ReadLine())
{
    case "a":
        Console.WriteLine($"Your result: {num1} + {num2} = " + (num1 + num2));
        break;
    case "s":
        Console.WriteLine($"Your result: {num1} - {num2} = " + (num1 - num2));
        break;
    case "m":
        Console.WriteLine($"Your result: {num1} * {num2} = " + (num1 * num2));
        break;
    case "d":
        Console.WriteLine($"Your result: {num1} / {num2} = " + (num1 / num2));
        break;
}

// Wait for the user to respond before closing.
Console.Write("Press any key to close the Calculator console app...");
Console.ReadKey();
```

- 1) Use the code above to produce a working calculator application
- 2) Test to make sure that each different calculation type (+,-,\*,/) work correctly

**\*Add a screenshot of the console for each calculation to your evidence log\***

## 5) Create a kilogram (kg) to pounds convertor (lb)

- 6) Create a simple conversion program using the skills developed in the calculator tutorial. You will need to investigate different numeric data types (e.g. NOT Int) to complete this task.

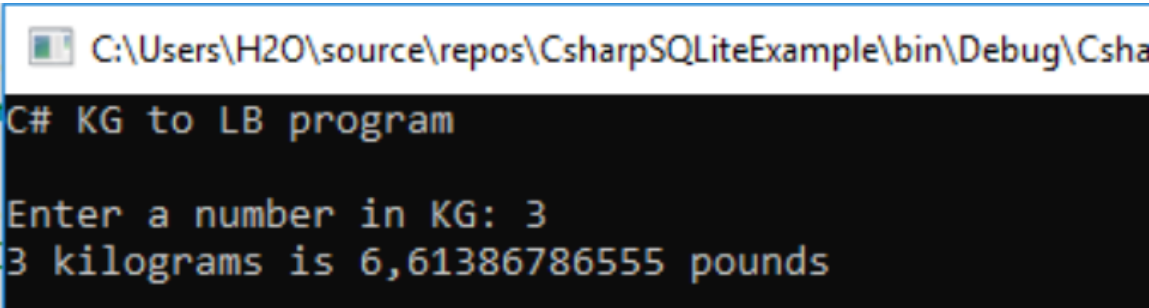
1 kilogram (kg) is equal to 2.20462262185 pounds (lbs).

1 kg = 2.20462262185 lb

Some of KG to LBS examples

Kilograms (kg)	Pounds (lb)
1 kg	2.205 lb
2 kg	4.409 lb
3 kg	6.614 lb
4 kg	8.818 lb
5 kg	11.023 lb

- 7) Your program should produce the following output



```
C:\Users\H2O\source\repos\CsharpSQLiteExample\bin\Debug\Csha
C# KG to LB program
Enter a number in KG: 3
3 kilograms is 6,61386786555 pounds
```

**\*Add a screenshot of the console to your evidence log\***

**\*Copy and paste your code to your evidence log\***

## 6) Binary to Decimal

- 1) Watch the video (see link below) covering how to convert a binary number to decimal number

<https://youtu.be/VLfITjd3IWA> (first 8 minutes)

- 2) On a sheet of lined paper complete the following conversion worksheet. You must clearly show your working for each question (include the number conversion line in each answer)

Write each binary number as a decimal number.

1. Binary = 1001  
Decimal =

2. Binary = 100000  
Decimal =

3. Binary = 1100101101  
Decimal =

4. Binary = 100001000  
Decimal =

5. Binary = 10101001  
Decimal =

6. Binary = 1010000100  
Decimal =

7. Binary = 1011010111  
Decimal =

8. Binary = 1010101111  
Decimal =

9. Binary = 1101111110100  
Decimal =

10. Binary = 110110101111  
Decimal =

**\*Take a clear photo of your completed answers (including working) and insert it into the evidence log\***

## 7) Binary Conversion (Fixed Point)

- 1) Watch the video (see link below) covering how to multiply two binary numbers

<https://youtu.be/VLfiTjd3IWA> (8 mins onwards)

- 2) On a sheet of lined paper complete the following conversion worksheet. You must clearly show your working for each question (include the number conversion line in each answer)

Q	Fixed Point	Denary	Q	Denary	Fixed Point
1	0011.0100		1	3.85	
2	0110.0111		2	4.62	
3	0010.1010		3	5.8	
4	0101.0110		4	7.6	
5	0111.1010		5	4.32	
6	0100.1000		6	6.6	
7	0111.0010		7	8.25	
8	0101.1100		8	7.38	
9	1010.1001		9	4.9	
10	1001.1001		10	3.26	

**\*Take a clear photo of your completed answers (including working) and insert it into the evidence log\***

## Induction Task Evidence Log

Task 1-2: A photo of your workstation's monitor with Visual Studio running <b>Insert photo here.</b>
Task 3: A console window opens that shows the sum of $42 + 119$ , which is <b>161</b> <b>Paste screenshot here.</b>
Task 4: Calculator (Addition) <b>Paste screenshot here.</b>
Task 4: Calculator (Subtraction) <b>Paste screenshot here.</b>
Task 4: Calculator (Multiplication) <b>Paste screenshot here.</b>
Task 4: Calculator (Division) <b>Paste screenshot here.</b>



Task 5: A console window opens that shows the conversion of 3kg to pounds

Paste screenshot here.

Task 5: Copy and paste the code used for task 5

Paste code here.

Task 6: Binary to Denary (Questions 1-10)

Insert a photo of your completed answers showing working here.

Task 7: Binary to Denary (Questions 1-10 x2)

Insert a photo of your completed answers showing working here.