



# Transition to A Level Computer Science Preparation Guide

To get a head start in *A Level Computer Science* it is important to learn Java programming as much as possible. By practicing **BEFORE** September you stand a **MUCH** greater chance of doing really well and achieving a higher grade. In fact, it is likely to help you so much that if you complete the tutorials and activities in this document it is estimated that you will achieve **1 to 2 grades higher** in your A Level!

*This document will take you only one to three hours to complete.*

Please email [rherbert@guildfordcounty.co.uk](mailto:rherbert@guildfordcounty.co.uk) or [meleftheriadi@guildfordcounty.co.uk](mailto:meleftheriadi@guildfordcounty.co.uk) if you need more information or guidance.

Lesson and homework tasks throughout the course are differentiated with **Beginner**, **Intermediate** and **Expert** tasks, meaning that if you have completed all the basics before starting the course, you will be well prepared for going straight for the **Intermediate** and **Expert** tasks!

During A Level a tracking sheet is used to show your progress in both programming and the theory topics. A leader-board in the classroom is used to display those students who have made the most progress.

Do not worry if you get stuck on a task or cannot get your code to work: just paste your code into the appropriate section below and we will discuss it in September. Remember, a computer is not intelligent, and can only do what you, the programmer, tell it to do – they need very precise and specific instructions!

Failing whilst programming, and using a trial and error approach, is the best way to learn! Success comes after failing, sometimes multiple times.

**It is the people who are willing to get things wrong and keep going who find the most success in life!**

**This full document has been emailed to you.**

There are many excellent tutorials online to help get you prepped and ready to do brilliantly.

At A Level we use the **Java** programming language and use **Net Beans** integrated development environment to help us write our **Java** programs.

To follow the links in this document, hold down the Ctrl key when you click on them or right click and do “Open Hyperlink”.

Work your way through the tutorial videos – pasting evidence of your programs next to each one. The more you do the better you’ll be! Feel free to do others that are not linked as well!

- I. The first thing you need to do is install **Netbeans** \* on your home computer. It is **FREE!**

Click [HERE](#) to download **Netbeans** and then follow the onscreen prompts to install it on your computer!

Java SE and NetBeans Cobundle (JDK 7u80 and NB 8.0.2)		
Product / File Description	File Size	Download
Linux x86	233.68 MB	<a href="#">jdk-7u80-nb-8_0_2-linux-i586.sh</a>
Linux x64	229.83 MB	<a href="#">jdk-7u80-nb-8_0_2-linux-x64.sh</a>
Mac OS X x64	301.5 MB	<a href="#">jdk-7u80-nb-8_0_2-macosx-x64.dmg</a>
Windows x86	243.83 MB	<a href="#">jdk-7u80-nb-8_0_2-windows-i586.exe</a>
Windows x64	246.65 MB	<a href="#">jdk-7u80-nb-8_0_2-windows-x64.exe</a>

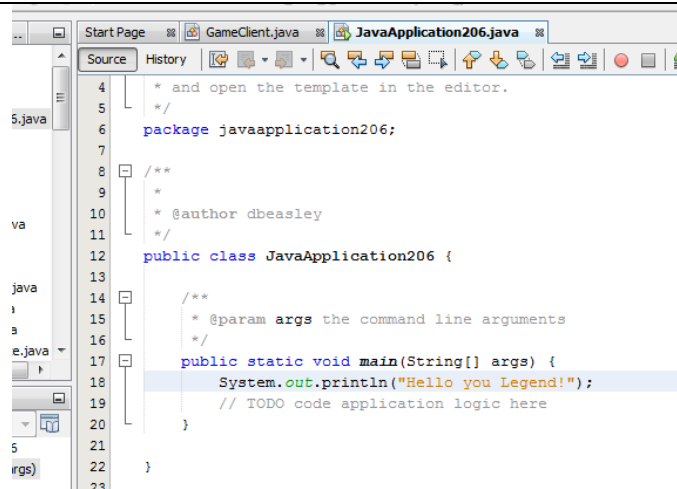
Once you have **Netbeans** installed it’s time to write our first program!



*\* if you have issues installing Netbeans (perhaps you are using an older MacOS) then perhaps install a more compatible IDE, e.g. Eclipse, BlueJ*

## 2. Creating your first Java program using Netbeans

**How to Output information to a user** – follow the Tutorial Video [HERE](#) and then Copy and Paste your program code for this below (or use the Print Screen and then Paste)



```
4  * and open the template in the editor.
5  */
6  package javaapplication206;
7
8  /**
9   *
10  * @author dbeasley
11  */
12  public class JavaApplication206 {
13
14  /**
15   * @param args the command line arguments
16   */
17  public static void main(String[] args) {
18      System.out.println("Hello you Legend!");
19      // TODO code application logic here
20  }
21
22  }
```

*Example:*

```
System.out.println("I am a legend");
```

**3. Using variables to store values in your program** – Click [HERE](#) to watch the tutorial on variables and then Copy and Paste your program code for this below

**4. Reading User Input to interact with your program** – Click [HERE](#) to watch the tutorial on reading User Input and then Copy and Paste your program code for this below

**5. Reading in numbers, using Decimals and performing calculations** – Click [HERE](#) to watch the tutorial and then Copy and Paste your program code for this below

**6. Maths operators** – Click [HERE](#) to watch the tutorial and then Copy and Paste your program code for this below

**7. Incrementing variables** – Click [HERE](#) to watch the tutorial and then Copy and Paste your program code for this below



**8. If statements** – Click [HERE](#) to watch the tutorial and then Copy and Paste your program code for this below

**9. Logical Operators** – Click [HERE](#) to watch the tutorial and then Copy and Paste your program code for this below

**10. Nesting Else If**– Click [HERE](#) to watch the tutorial and then Copy and Paste your program code for this below

**11. While Loop**– Click [HERE](#) to watch the tutorial and then Copy and Paste your program code for this below

## **Programming Challenges!**

Following instructions is one thing – but can you show your problem solving skills to create a program for each of the challenges below?

For some of them you may need to view additional tutorials or use Google for help (For each one you complete or attempt, paste your code below the relevant challenge)

1. Create a program that will output the subjects you are studying in Year 12
2. Create a program that will ask the user what colour is their favourite and then print out their choice. E.g. “What is your favourite Colour” ->Black “I see your favourite colour is Black”.
3. Create a program that will ask someone how old they are and then tell them how old they will be in 10 years’ time.
4. Create a program that will ask the user a maths question and then will tell them if they got it right or wrong.
5. Create a program that will ask them repeatedly if they are feeling annoyed. It will keep going until they type “yes”.
6. Create a program that will give the user a quiz. Each question they get right should make their score increase. At the end of the quiz tell them their score! At the end of the quiz ask them if they want to take the quiz again

**If you’ve completed or even attempted all the above, congratulations you are in great shape and well prepared for your Computer Science A Level!**