Computer Science
Year 11 into year 12 summer independent work
Deadline – first lesson in September 2025

The summer independent work is an opportunity for you to showcase your ability as a programmer, and also your ability to think ahead and plan your algorithms.

The compulsory section of work comes in two parts, firstly you will create some programs in Python using learnpython.com, then you will apply the knowledge and skills you have developed to complete five programming challenges. There are then some additional optional activities you might choose to complete.

*Evidence required – Word or PowerPoint evidence document with all of the solutions **clearly labelled.***

Compulsory – must do!

Python programming
1 – Hello world
https://www.learnpython.org/en/Hello%2C_World%21

Exercise:

1. Have a go at each of the warm up tasks, be sure to read them carefully
2. Paste your code for the final exercise on your evidence document, and label clearly

2 – Variables and types
https://www.learnpython.org/en/Variables_and_Types

Exercise:

1. Have a go at each of the warm up tasks, be sure to read them carefully
2. Paste your code for the final exercise on your evidence document, and label clearly

3 – Lists
https://www.learnpython.org/en/Lists

Exercise:

1. Have a go at each of the warm up tasks, be sure to read them carefully
2. Paste your code for the final exercise on your evidence document, and label clearly

4 – String formatting
https://www.learnpython.org/en/String_Formatting

Exercise:

1. Have a go at each of the warm up tasks, be sure to read them carefully

2. Paste your code for the final exercise on your evidence document, and label clearly

5 – String operations
https://www.learnpython.org/en/Basic_String_Operations

Exercise:

1. Have a go at each of the warm up tasks, be sure to read them carefully

2. Paste your code for the final exercise on your evidence document, and label clearly

6 – Conditions
https://www.learnpython.org/en/Conditions

Exercise:

1. Have a go at each of the warm up tasks, be sure to read them carefully

2. Paste your code for the final exercise on your evidence document, and label clearly

7 - Loops
https://www.learnpython.org/en/Loops

Exercise:

1. Have a go at each of the warm up tasks, be sure to read them carefully

2. Paste your code for the final exercise on your evidence document, and label clearly

8 – Reusable code
https://www.learnpython.org/en/Functions

Exercise:

1. Have a go at each of the warm up tasks, be sure to read them carefully

2. Paste your code for the final exercise on your evidence document, and label clearly

Once you have completed the tutorials, you should now have the knowledge and skills to complete the following challenges.

You can use an online editor such as Replit.

Or you will need to install the Python interpreter, and could use its built in IDE IDLE.

Challenge 1 – Number table

Number Table

Write a program that takes a symbol (+,-,* or /) and a natural number (>0) and makes a table like below for the operation from 0 to n For this example the user has entered "+ 4":

```
+ |  0  1  2  3  4

------------------

0 |  0  1  2  3  4

1 |  1  2  3  4  5

2 |  2  3  4  5  6

3 |  3  4  5  6  7

4 |  4  5  6  7  8
```

Paste your code for this challenge on your evidence document, and label clearly.

Challenge 2 – Logic gate

Logic gate

Write a program that will give the students the answer to logic gate questions For example:

Enter logic gate : OR

Enter first input : 1

Enter second input : 0

Result = 1

It should work for the logic gates OR, AND, XOR, NAND and NOR

Paste your code for this challenge on your evidence document, and label clearly.

Challenge 3 – What's the day?

What's the day?

Design a program to take 3 inputs, one for day, one for month and one for year. Get your program to validate if this is an actual day, and, if it is, output the day of the week it is!

Hint: How do leap years affect this program?

Paste your code for this challenge on your evidence document, and label clearly.

Challenge 4 – Two player dice game

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.

Develop 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.

Paste your code for this challenge on your evidence document, and label clearly.

Challenge 5 – Zork style adventure game

Exercise: Play the text adventure game Zork

That's right, I am asking you to play a game, and that forms part of your Summer Independent Learning!

At the time of writing this link worked to a web version of Zork
http://textadventures.co.uk/games/view/5zyoqrsugeopel3ffhz_vq/zork

If it doesn't, then Google "Text Adventure Zork Online" and you should be able to find a link.

There is method to my madness, playing a cutting edge game like this (for 1977ish) allows you to think about all the skills a programmer needs:

- Use of variables
- Inputs
- Outputs
- Lists
- Operators
- Formatting strings
- Conditions and branching
- Loops or iteration
- Functions
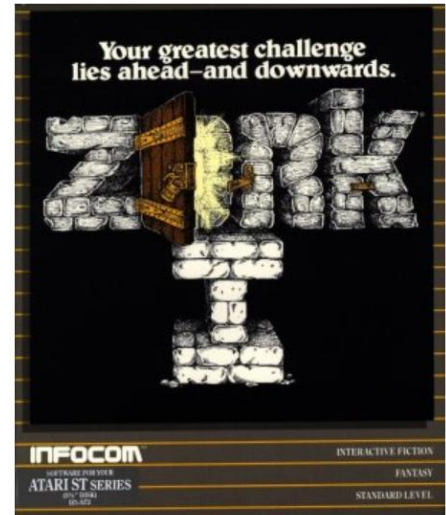
As you play think about what must be happening under the hood, how do you collect items, have battles, have choices, get random responses, move around?

If you get stuck, try the help guide here: http://www.eristic.net/games/infocom/zork1.html

To think about:

1. Who are your characters?
2. Are they playable?
3. Who are the non playable characters?
4. What skills or weapons can you pick up along the way?
5. What items can you carry and how many?
6. How do you win?
7. Will you have health or lives?
8. How do you manoeuvre around?

Paste your code for this challenge on your evidence document, and label clearly.

Help 1: Tech with Tim

https://youtu.be/DEcFCn2ubSg

This is a basic introduction to a text adventure game on YouTube – a good place to get you started.

Help 2: Invent with Python

https://inventwithpython.com/invent4thed/chapter5.html

This game was has a lot of similar features to Zork and will help if you read the explanation and code.

There is so much help out there – a quick search on a search engine should provide you with lots of tutorials for a text adventure in Python. I would also strongly recommend YouTube.

Optional Activities

Task 16 (optional) – Who is John Von Neumann and what was his impact on computing?
- https://en.wikipedia.org/wiki/John_von_Neumann

- https://www.famousscientists.org/john-vonneumann/

- https://www.geeksforgeeks.org/computerorganization-von-neumann-architecture/

- https://youtu.be/QhBvuW-kCbM

- https://youtu.be/Ml3-kVYLNr8

Task 17 (optional) – Who is Alan Turing and what was his impact on computing?
- https://en.wikipedia.org/wiki/Alan_Turing

- https://www.bbc.co.uk/teach/alan-turing-creator-ofmodern-computing/zhwp7nb

- https://www.youtube.com/watch?v=7TycxwFmdB0

- https://youtu.be/XQMXOCTG2Gw

Task 18 (optional) – Who is Ada Lovelace and what was her impact on computing?
- https://en.wikipedia.org/wiki/Ada_Lovelace

- https://www.britannica.com/biography/Ada-Lovelace

- https://www.famousscientists.org/ada-lovelace/

- https://youtu.be/1QQ3gWmd20s

- https://youtu.be/wnHHzBY1SPQ

Task 19 (optional) – Who is Tim Berners-Lee and what was his impact on computing?
- https://en.wikipedia.org/wiki/Tim_Berners-Lee

- http://www.computinghistory.org.uk/det/2489/TimBerners-Lee/

- https://youtu.be/OM6XIICm_qo

- https://youtu.be/ykCPFUIasFk

Task 20 (optional) – Past papers
In this task, you get the chance to look at some past papers for A Level Computer Science. It is a great
opportunity to see the format of the exam and the style of questions. Can you attempt any questions already through existing knowledge?

https://www.ocr.org.uk/Images/666846-question-paper-computer-systems.pdf

https://www.ocr.org.uk/Images/666847-question-paper-algorithms-and-programming.pdf

Great resources to help with A Level Computer Science:

https://adacomputerscience.org/

https://replit.com

https://www.physicsandmathstutor.com/computer-science-revision/a-level-ocr/

https://www.amazon.co.uk/OCR-AS-Level-Computer-Science/dp/1910523054/ref=sr_1_1?crid=2SGXGISJRUSCT&keywords=ocr+a+level+computer+science&qid=1655463471&sprefix=ocr+a+level+computer+science%2Caps%2C66&sr=8-1

https://www.youtube.com/channel/UC0HzEBLIJxlrwBAHJ5S9JQg

https://www.youtube.com/playlist?list=PLH2l6uzC4UEW0s7-KewFLBC1D0l6XRfye

https://www.youtube.com/user/Computerphile

https://www.youtube.com/channel/UCsBxhDfwURg-vQASN2ZeHwg

Please ensure your SIL is complete in time for your first lesson in September.

You will submit it electronically, there is no need to print.