



Computer Science

Year 11 into year 12 summer independent work

Deadline – first lesson in September 2022

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.

It 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.

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

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 will need to install the [Python interpreter](#), and could use its built in IDE IDLE.

I recommend using a Python IDE like [pycharm](#), [VS code](#) or [sublime text](#) to produce the code so that you can benefit from the built in functions. You are not limited to the IDE's on this list, you can use whatever you are used to.

You can also use an online editor such as [replit](#).



Challenge 1 – Heads or tails

Heads or tails

Create a program that asks the user how many times they would like to simulate tossing a coin. It should then generate either a 'head' or 'tails' randomly, for the specified number of times. It should print out at the end:

1. How many coin tosses it simulated
2. How many heads in total
3. How many tails in total
4. The best head streak – best run of heads in a row without any tails
5. The best tails streak – best run of tails in a row without any heads

Below are example outputs to help you work out how to program the solution:

```
How many coin tosses would you like to simulate?
-> 10
Result 1 : tails
Result 2 : heads
Result 3 : heads
Result 4 : tails
Result 5 : tails
Result 6 : heads
Result 7 : tails
Result 8 : heads
Result 9 : tails
Result 10 : tails
The total count of heads: 4
The best streak for heads: 2
The total count of tails: 6
The best streak for tails: 2
Would you like to simulate again?(y/n)
->
```

```
Would you like to simulate again?(y/n)
-> y
How many coin tosses would you like to simulate?
-> 20
Result 1 : tails
Result 2 : tails
Result 3 : heads
Result 4 : tails
Result 5 : heads
Result 6 : heads
Result 7 : heads
Result 8 : tails
Result 9 : heads
Result 10 : heads
Result 11 : tails
Result 12 : heads
Result 13 : tails
Result 14 : heads
Result 15 : heads
Result 16 : tails
Result 17 : heads
Result 18 : heads
Result 19 : heads
Result 20 : tails
The total count of heads: 12
The best streak for heads: 3
The total count of tails: 8
The best streak for tails: 2
Would you like to simulate again?(y/n)
->
```

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



Challenge 2 – 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 the final exercise on your evidence document, and label clearly



Challenge 3 – 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 the final exercise on your evidence document, and label clearly



Challenge 4 – 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 the final exercise 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!

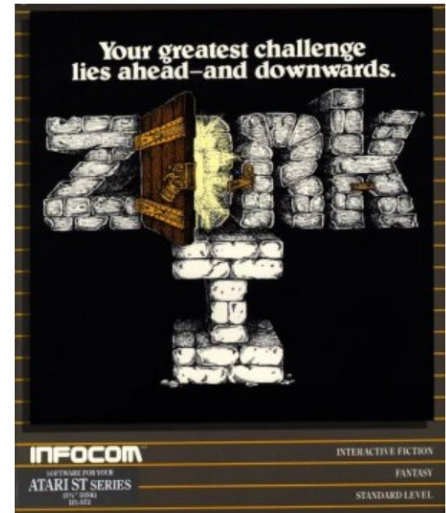
At the time of writing this link worked to a web version of Zork

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 the final exercise 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.

My email is mark.ibbertson@nclt.ac.uk if you need to contact me.

Head of School – Maths, IT and Computing
Disadvantaged and Covid catch-up – Strategy and quality manager
New College Doncaster

T: 01302 976777 ext 437

W: www.ncdoncaster.ac.uk

A: Hurst Lane, Auckley, Doncaster, DN9 3HG

NCLT fully support our staff to achieve a healthy work-life balance.

It is therefore not an expectation from the Trust that staff respond to emails sent outside of normal working hours.

