

Computer Science Transition Tasks



Computer Science is a creative, practical, reasoning, and problem-solving subject applied to the real world.

The programmer gets the chance to create something from nothing, use logic to turn programming concepts into code that a computer can run, and, when things don't work as expected, use problem solving to figure out what has gone wrong.

Programming is a fun, sometimes challenging (and occasionally frustrating) activity, and the skills learned from it can be useful both in school and at work . . . even if your career has nothing to do with computers.

And, if nothing else, programming is a great way to spend an afternoon when the weather outside is rubbish.

TASK SET

The tasks below will help you get prepared with some of the things you will be learning on the course. Why do we do this? Well, every year we get students who start the course thinking they are good at using computers but have used them mostly in an ICT way rather than a computing way. So treat this summer task as a crash course so you know the basics. And if you have successfully completed GCSE Computer Science, it also gives you a chance to brush up on skills you may have forgotten as soon as you walked out of the exam hall, and to learn new things you didn't know.

What you will need

- A Computer with an operating system.
 1. This can be Windows, OS X, or Linux
 2. Access to the internet (for research)
 3. Python (latest version is 3.3)

Task 1 – About You

1. Why have you chosen the study Computer Science at A Level?
2. What do you aim to achieve from the course?
3. What skills are you aiming to develop?
4. What other A Levels have you chosen and why?
5. Describe what you were like as a GCSE student. (Answer honestly! It will help me make your learning more appropriate)
6. Is there anything I need to know about you? (Regarding your learning!)

Computer Science Transition Tasks



Task 2: Truth Tables – Task Sheet

1. Write the truth tables for the expressions
NOT (A AND B)

and ((NOT A) OR (NOT B))

2. What do you notice about these tables?

3. Design (using algorithms) and create a program to output the value of a after the statement
IF (a < b) OR (b < c) THEN a = b
has been executed.

Computer Science Transition Tasks



Data types, data structures and algorithms

Converting between denary, binary and hex

No.	Denary	Binary	Hex	Binary value plus 00011110
1	1			
2	5			
3	10			
4	22			
5	40			
6	77			
7	91			
8	121			
9	144			
10	168			
11	170			
12	200			
13	211			

Create a program that analyses a passage of text from a file and then counts:

- How many words
- The average length of a word
- How many times each word occurs
- How many words start with each letter of the alphabet?

The aim of this exercise is to test your ability to develop algorithms.

Computer Science Transition Tasks



Extension: Problem Solving

RPG character simulator

Planet of Fightcraft wants you to build character classes for their new game.

Each character will have the following things:

- Name
- Type (Barbarian, Elf, Wizard, Dragon, Knight)
- Health
- Power
- Special attack power
- Speed
-

All characters start with 100 health

Different creatures have different power ratings (B: 70, E: 30, W: 50, D: 90, K: 60)

Different creatures have different special attack power ratings (B: 20, E: 60, W: 70, D: 40, K: 10)

Different creatures have different speed ratings (B: 50, E: 10, W: 30, D: 50, K: 60)

Tasks

1. Generate a random name: en-da-fu and el-kar-tuk could be names, so you could make a name generator which sticks together three syllables from 'word banks'
2. Create the generic character class. Test to see if you can create multiple characters
3. Create subclasses corresponding to different types of creature (B, E, W, D & K)
4. Make a program that randomly generates 10 of these creatures to add into a list
5. Make a method in the character class that enables printing out of each character's stats to the console
6. Create a menu system that lets you add and delete characters and print out the list until you are happy with the team
7. Create methods to let you edit any character's stats and add this to your menu system
8. Create a way to save your team to a file and load it up again if needed