IT JOBS 2018

Cardinal Newman
CATHOLIC SCHOOL

St Alban
Catholic
ACADEMIES TRUST

In the past 12 months, companies
have hired workers in these areas:*
46% IT SUPPORT
38% DEVELOPERS
29% IT MANAGEMENT
29% NETWORK ADMINS

IT professionals
say the most job
creation will come
from these areas:
NETWORK SECURITY
DATA ANALYSIS

Welcome to A-Level Computer Science. Choosing Computer Science (OCR exam board) places you at the cutting edge of opportunities. You will be the most sought after person. Our course consists of 2 exams (40% each exam) and a project (20%). Completing our transition task will help us understand you programming skills. Please complete by 1st week in September.

TRANSITION TASK

More than 50% of A Level is programming so it is imperative that you continue to develop your programming skills. The following are divided into compulsory and optional. The optional tasks are more complex programming questions which use structures that have been taught in GCSE. It is VITAL that you at least try them and please produce evidence that you have tried. It is NOT about getting the optional tasks correct BUT it's about trying.

Compulsory tasks

The following tasks are a recap of GCSE programming tasks. These are compulsory.

1. Generate a random number between 1 and 9 (including 1 and 9). Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right.
2. Make a two-player Rock-Paper-Scissors game. (*Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game*). Remember the rules:
   o Rock beats scissors
   o Scissors beats paper
   o Paper beats rock
3. Write a program where the you ask the user to input a number and the program will determine whether the number entered is prime or not. (For those who have forgotten, a prime number is a number that has no divisors)
4. Write a program that takes a list of numbers (for example, a = [5, 10, 15, 20, 25]) and makes a new list of only the first and last elements of the given list.
5. Implement a function that takes as input three variables, and returns the largest of the three. Do this without using the Python max () function!
6. Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. Make sure to ask the user to enter the number of numbers in the sequence to generate. (*Hint: The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, …*)

1. Write a program (using functions!) that asks the user for a long string containing multiple words. Print back to the user the same string, except with the words in backwards order. For example, say I type the string:

   My name is Michele

   Then I would see the string:

   Michele is name My

   shown back to me.

2. Create a program that will play the "cows and bulls" game with the user. The game works like this:

   Randomly generate a 4-digit number. Ask the user to guess a 4-digit number. For every digit that the user guessed correctly *in the correct place*, they have a "cow". For every digit the user guessed correctly *in the wrong place* is a "bull." Every time the user makes a guess, tell them how many "cows" and "bulls" they have. Once the user guesses the correct number, the game is over. Keep track of the number of guesses the user makes throughout the game and tell the user at the end.

   Say the number generated by the computer is 1038. An example interaction could look like this:

   Welcome to the Cows and Bulls Game!

   Enter a number:

   >>> 1234

   2 cows, 0 bulls

   >>> 1206

   1 cow, 1 bull

   ...

   Until the user guesses the number.

3. For this exercise, we will keep track of when our friend's birthdays are, and be able to find that information based on their name. Create a dictionary (in your file) of names and birthdays. When you run your program it should ask the user to enter a name, and return the birthday of that person back to them. The interaction should look something like this:

> >>> Welcome to the birthday dictionary. We know the birthdays of:
>
> Albert Einstein
>
> Benjamin Franklin
>
> Ada Lovelace
>
> >>> Who's birthday do you want to look up?
>
> Benjamin Franklin
>
> >>> Benjamin Franklin's birthday is 01/17/1706.

4. In this exercise, the task is to write a function that picks a random word from a list of words from the SOWPODS dictionary. Download this file and save it in the same directory as your Python code. This file is Peter Norvig's compilation of the dictionary of words used in professional Scrabble tournaments. Each line in the file contains a single word.

Wider reading and preparation for A Level

https://www.youtube.com/watch?v=FpMNs7H24X0

http://www.cs4fn.org/

http://www.tnmoc.org/

http://pcpro.com

http://www.theregister.co.uk/