

## Preparatory work for Y11s who are to transition to A-level Computer Science

### What to expect:

- in-depth study of Computer Science theory
- develop problem solving skills
- be able to use Python programming language to implement solutions to a range of problems ranging from simple to very complex

### Familiar with yourself with the curriculum:

- have a general look at this link <https://www.bournetocode.com/> where all notes of the course is published under the A-level area

### To prepare yourself for a successful A-level course:

- if you did not take Computer Science for GCSE
  - for some basics of Computer Science theory, please follow this course <https://codehs.com/go/99B54>
  - for Python programming, please follow this course <https://codehs.com/go/7FD64>
  - create an account when asked and we will be able to track your progress
- if you have taken GCSE Computer Science
  - please following this course to brush up your Python skills <https://codehs.com/go/7FD64>
  - create an account when asked and we will be able to track your progress
  - when finished with the above course OR you are confident that the above course is too simple to begin with, you can proceed to solve the following programming problems in the order of your choice.
  - To work on the programming tasks, you need Python software or online Python coding site. Here are some choices for you:
    - Install Python IDLE from this site <https://www.python.org/downloads/> for windows, linux or mac.
    - If you don't want to install software, here is an online Python environment <https://repl.it/languages/python3> You can sign up there and save your code
    - Each task is a skeleton code for you to begin work with by copying and paste into your own software for editing and testing

## Problem 1:

```
# make sure you use the following data for your code
a = "a"
b = "12"
c = True
d = 12

# Task 1:
# create a function if you know how that will:
# check a variable for data type and output the type for each
# test your function using the correct variables

# Task 2: improve your function so that:
# when a string is given, check that if it is alphabet or all number and output
the result
# test your function using the correct variables

# Task 3: improve your function so that:
# when a string is given, check that if it is a character and output the result.
# test your function using the correct variable

# Task 4: if you have done the above easily, here is a fun one to try:
# Write a magic 8 ball app that will:
# 1. give players at least one of random advices from 5
# 2. allow users to ask a question
```

## Problem 2:

```
# Task 1: create a function that will convert a given decimal to binary

def decimalToBinary(n):
    # your code here and remove the pass
    pass

# Task 2: add a feature to your solution so a user can input a decimal number
# Then you will call the function and convert to binary

# Task 3: improve your code so that only a valid decimal value will be accepted
# Keep asking users for a valid decimal value if an invalid value is inputted
```

### Problem 3:

```
# data validation exercises
# add your code to validate all inputs or random picks

# REMEMBER:
# all input by users through key board typing are string data type

''' task 1:
    splitting the bill evenly after the input is given
    what is the expected data type for totalBill?
    what is the expected data type for numOfPeople?
'''

totalBill = input("Input the total bill ")
numOfPeople = input("Input the number of people in your group: ")

''' task 2:
    pick a random item from a list
    the list contains a mixture of data types
    only number data type should be picked
'''
myList=['3', 'Peas', 5, 'Fish', '24', 'Chips']

''' task 3:
    pick a random location to place a letter B in the 2D array
    B can only be place on the grid where there is no x
'''

my2DArray=[['x', 'x', '-', 'x', 'x', '-', '-', '-'],
            ['- ', '- ', '- ', 'x', '- ', 'x', '- ', '- '],
            ['- ', 'x', '-', '-', '-', '-', '-', '-'],
            ['x', 'x', '-', '-', 'x', '-', '-', '-'],
            ['- ', '- ', '-', 'x', '-', '-', '-', '-'],
            ['- ', 'x', '-', '-', '-', 'x', '-', '-'],
            ['- ', '- ', 'x', 'x', '-', '-', '-', '-'],
            ['- ', '- ', '-', '-', '-', 'x', '-', '-']]
```

#### Problem 4:

```
myList =[11,88,66,44,55,33,77,22,99]
# output the number of items in the list

# display each item without using loop

# display each item using loop
# display the index position of any given item

# split the list into two lists: [11, 88, 66, 44], and [55,33,77,22,99]

# add the item 100 to the end of myList

# merge the list [200, 300, 500] to myList

# remove all items in the list myList
```

## Problem 5:

```
# A list for the task 1 - task 5
myList = [11, 88, 66, 44, 55, 33, 77, 22, 99]

# Task 1: using for loop to add all numbers in the list together

# Task 2: using while loop to add all numbers together

# Task 3: using for loop or while loop to add only numbers in the odd positions

# task 4: reverse the order of the list without using any built-in functions

# Task 5: sort the list in descending order without using any built-in functions

# Task 6: check if the list is symmetrical.
# Test it by using both symmetrical and non-symmetrical lists below:
# Test data 1: myList=[11,22,33,44,44,33,22,11]
# Test data 2: myList=[11,22,33,44,33,22,11]
# Test data 3: myList=[11,33,33,44,33,22,11]
```

## Problem 6:

```
myList = ['eggs', 'bacon', 'beans', 'toasts', 'tomatos', 'milk', 'pineapple']

# Task 1: output all the item in the following format:
'''
1. Eggs
2. Bacon
3. Beans
4. Toasts
6. Tomatos
7. Milk
8. Pineapple
'''

# task 2: output only the items with six or more letters

# task 3: output items only with two or more vowels

# task 4: count how many the occurances of each alphabet assuming lower case only

# task 5: remove items that have the letter "o" in it without using another list

# task 6: sort the new list with no items with "o" using Bubble sort
```

If you are really up for some challenges, try to implement solutions for the following problems

Exercises	Skills needed	I have learned	I have coded it and tested it works(yes/no)
<p><b>Pangram</b></p> <p>"The quick brown fox jumps over the lazy dog"; note how all 26 English-language letters are used in the sentence. This is called a <b>pangram</b>.</p> <p>implement a program that takes a series of strings (one per line) and prints either True or False if the input string is a pangram.</p>	<p>Input Loops (for or while) Conditional List operations</p>		
<p><b>Fruit Machine</b></p> <p>Write a program to simulate a Fruit Machine that displays three symbols at random from Snowman, Heart, Spade, Clover, Star, Skull.</p> <p>Define a function that will accepts two parameters, rolled results and current credit, and returns the money left after process the rolled results based on the following rules: The player starts with £1 credit, with each go costing 20p. If the Fruit Machine "rolls" two of the same symbol, the user wins 50p. The player wins £1 for three of the same and £5 for 3 Hearts. The player loses £1 if two skulls are rolled and all of his/her money if three skulls are rolled. The player can choose to quit with the winnings after each roll or keep playing until there is no money left.</p>	<p>Random module Loops (for or while) Conditional Dictionary List operations <b>Function</b> Unicode for the symbols: "\u2620" – skull "\u2661" – heart "\u2618" – clover "\u2606" – star "\u2603" – snowman "\u2667"-spade</p>		
<p><b>Text file utility</b></p> <p>Implement a program that read a .CSV file of records, sort them by one field, and then write them back to the file. Allow the user to choose various sorting orders based on a chosen field.</p>	<p>File input File output Loops (for or while)</p>		
<p><b>Hack the pin code (Extension Challenge)</b></p> <p>A thief has managed to find out the four digits for an online PIN code, but doesn't know the correct sequence needed to hack into the account.</p> <p>Design and write a program that displays all the possible combinations for any four numerical digits entered by the user. The program should avoid displaying the same combination more than once.</p>	<p>input loops (for or while) list operations conditional <b>function or recursive function</b></p>		