1. Instructions & Introduction

1.1 Preface

- The AS level Computer Science paper 2 asks you to write some very logical and, of course, interesting programs. A lot of you will fear and pray for the easiest to come – but when you think about it, it isn't very hard to ace this paper.
- So this is a guide for you to ace the paper 2 of Computer Science. It will walk you through what you should learn, small program questions that will help you in the exam and also will refer to many parts in the practical paper of AS Computer Science.
- You will have to learn pseudo code. It ALWAYS COMES and it can't be avoided. But this, with minimal practice, will come along the way.

1.2 Language Decision

- You have a choice between VB.NET, Pascal, or Python. VB is a nice choice, Windows uses it for most of their apps and it is widely used today. Pascal was in the time of C. So it is quite old and syntax will be lengthy. Python is a new high level language. Whatever you choose, do NOT choose the language because it's the "trend" or because everybody is insisting on it. Chose what you think is rather comfortable and fun for you. There is no partial grading in what language you choose. For instance, I'd prefer a language like Pascal because it's not only fun, but also indirectly teaches me the syntax that is similar to Java.

1.3 The Pre-Release Material

- For the AS level paper, the board does send you a pre-release paper a few months before the exam. The best approach to this:

1. Read the instruction on the first page and see the appendix at the last page, see if you notice anything new and familiar
2. Close that pre-release material and don't open it for the next month or so (this is assuming you are reading this when you got the material) If you are reading this when the exam is around the corner – then it's better that you finish it by the week before the test so you have time to go through it
3. So if this material is new, sharpen your programming skills to the end, know syntax at your fingertips, by heart important methods/ functions you can make use of
4. Then start sitting and working on your material – it'll just come by really easy for you. It's better to solidly do it a month before the test, where it's fresh in mind, then waste time trying to solve the material for months not knowing what you need for it

But you need questions, or puzzles, to test yourself. And that's why this guide has been made.

2. The 'Must-Know' Areas of Paper 2

- Once you know what language you have selected there are many foundational things you need to know. This is the list
    1. The data types: what is valid and what isn't?
    2. Arithmetic operators
    3. Logical operators
    4. Iteration
    5. Arrays and Lists
    6. Reading and writing on files
- This guide will not go in detail of each topic (we have amazing note for it, so check that out). It is assumed that you know all of these areas very well. We suggest that before you continue you with the guide, familiarize with yourself in these topics.

2.1 Program Exercises

- CS may be a wall to many of you, and that is because you don't have the CS *mindset*. Your thoughts need to flow in a different way. Every problem can be solved with a program. But to solve the problem, you need logic. It's a puzzle. Just as you crack a Sudoku, you can crack a problem. Don't worry. With practice, you can create a powerful *mindset*, just like any other top-notch programmer can.
- We've written these programs after analyzing the past paper program questions. These are small questions that can help you save time.
- The difficulty will rise as you solve questions. Don't fear if you can't solve the question, take a break (trust me on this one. You shouldn't burn out your mind all the time). And if you still don't get it, then follow these steps:

1. Ask yourself: What is the question asking me to do?
2. Once you have that, then see whether there are built-in functions that you can use
3. What variables do I need ad what will it store? Do they need to be changing values? Global values? (Tip: Remember you need to user proper variables that a reader can understand. If the variable is used for a counter, name it "counter" and not "var1")
4. Draw a data flow diagram – see where all the data needs to go and what output you need and o you should achieve that output

3. Exercises

- Directions: Write each program in pseudo code then your language you have chosen for the exam. Don't see the answers till you have tried the question first.

1. Write a program where a user can enter they're name and prints the name on the screen.
2. Write program where you have to swap the values on two variables that have been given as an input to the system. (if a =6 and b =0 then a =0 and b=6)
3. Write a program that evaluates an input and sees whether it is equal to, less than, or greater than a constant value of your choice
4. Write a program that converts the input from al CAPS to all lowercase.
5. Find a number that counts the number of values in an array
6. Write a program that generates 6 random numbers between 1-50 and prints on screen.
7. A program that prints only even numbers
    a. Odd numbers
    b. Prime numbers
8. A user inputs numbers. Convert that words from 0-10. (example: input is 1. Then output is one)
9. Two teams are playing Basketball. Write a program that inputs the names of both teams, their score and who won by the difference of the score.
10. For the program in question 4, write a function where it will detect whether the number has been previously drawn after each draw. (You can use anything you need to store the 6 numbers)
11. Write a program that makes a pyramid of the word of input by sub stringing.
12. There are numbers 1 to 100. But one number is missing. Write a function that finds that number
13. Enter in 10 numbers of your choice in an array in any order. Now sort the array using bubble sort
14. In the same array created (question ##), write a function that searches for a particular value and will output the index it is located
15. Enter a word in a program. At a particular index of the string, the character needs to display of the ASCII equivalent. Write a program to do this instruction
16. Write a program that gives the output of the grade you receive based on the marks of a subject. Hence, input all the subject grade and give out the average grade.
17. A function is required to convert currency from USD to any currency of your choice. Write a program to do so.
18. A function acts like a text bot. Example (if you type in hello, the system should say hello back) Do this for 3 different commands and put it in one function.
19. Let the user give any number and the system should print out the multiplication table
20. A company needs a program that automatically gives the pay scheme that the end of the month for three types of workers. 1) Managers - $150/day 2) Driver - $100/day 3) Supervisor $90/day. If everybody in the three categories didn't take a leave and it was a September that month, calculate the pay schemes for each type of person.