COMPUTER SCIENCE A LEVEL

PREPARATION WORK

St Albans Girls’ School



Summer 2021

The transition from working at GCSE standard to an A-Level is significant, including emphasis on technical content, extended answers and independent research. This summer work is designed to allow you to practice some of these skills, building on the work that you may have covered at GCSE.

The questions are designed to go beyond GCSE standard and prepare you for A-Level study. Some questions are chosen to give you a chance to extend both your thinking and writing skills and to demonstrate your creativity solving problems.

Please complete all the questions in Word or Google Docs and send them to [mathsbs@stags.herts.sch.uk](mailto:mathsbs@stags.herts.sch.uk) before your first lesson or alternatively hand them in during your first Computer Science lesson.

**WIDER COMPUTING ISSUES AND INTEGRATED SOLUTIONS**

These questions require you to use your technical knowledge in context. Reference any sources that you use to help you.

**1.** Create a timeline showing the history of computing, including any key discoveries or inventions. Extend youR timeline to show how you think Computer Science might develop over the next 50 years.

**2**. Compare the Xbox ONE, PS4 Pro and PC as gaming platforms. You must use as much technical detail as possible and reference any evidence presented.

**3.** Discuss the benefits and limitations of Virtual Reality a. In business contexts, such as medicine b. As a gaming tool c. As an extension to social media

**SYSTEMS ARCHITECTURE**

**1.** Produce an annotated diagram showing how the CPU processes data. This should include:

a. The purpose of the CPU

b. Common CPU components and their function

i. Arithmetic and Logic Unit (ALU)

ii. Control Unit (CU)

iii. Cache

iv. Registers

1. Memory Address Register (MAR)

2. Memory Data Register (MDR)

3. Program Counter

4. Accumulator

c. Reference to the fetch-execute cycle

**2.** Discuss, with examples, how the performance of a CPU can be improved, including:

a. Increasing the clock speed

b. Increasing the cache size

c. Increasing the number of processing cores

**MEMORY**

**1.** Compare RAM and ROM

**2.** Explain the need for virtual memory in a computer system

**3.** Describe the characteristics of flash memory

**NETWORKS**

**1.** Explain the similarities and differences between:

a. A LAN and a WAN

b. Client-server and peer-to-peer networks

**2.** Explain the difference between the Internet and the World Wide Web

**3.** Describe the factors that affect network performance, and explain how network performance can be improved

**4.** Draw three different network topologies:

a. Label all the components required to create each network

b. Explain the purpose of each component in the network, including

i. Wireless Access Points

ii. Routers

iii. Switches

iv. Network Interface Cards

v. Transmission media, such as Ethernet Cables

**COMPUTATIONAL LOGIC AND CALCULATIONS**

**1.** Complete the truth tables for the following expressions:

1. A AND (B OR C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **B OR C** | **A AND (B OR C)** |
| 0 | 0 |  |  |  |
| 0 | 0 |  |  |  |
| 0 | 1 |  |  |  |
| 0 | 1 |  |  |  |
| 1 | 0 |  |  |  |
| 1 | 0 |  |  |  |
| 1 | 1 |  |  |  |
| 1 | 1 |  |  |  |

1. (NOT A) OR (NOT B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Draw a circuit to represent each expression

**2.** Calculate each of the following:

a. 13 MOD 2

b. 16 MOD 6

c. 15 MOD 3

d. 7 MOD 8

e. 13 DIV 2

f. 16 DIV 6

g. 15 DIV 3

h. 7 DIV 8

i. 2^0

j. 2^7

k. 2^8

l. 2^10

**3.** Convert the following into the units given:

a. 4 bytes = bits

b. 1 TB = bytes

c. 80 kB = GB

d. 40 MB = nibbles

**4.** Complete the table, converting between binary, hexadecimal and denary as required:

|  |  |  |
| --- | --- | --- |
| **Binary** | **Hex** | **Denary** |
| 0010 1010 |  |  |
|  | 0B |  |
|  |  | 255 |
| 0110 0111 |  |  |
|  | F5 |  |
|  |  | 48 |
|  | CD |  |

**5**. Complete the following calculations:

a. 0110 0011 + 0011 0001

b. 1010 0110 + 1100 1111

c. 0110 0011 << 2 (bit shift left two places)

**PROGRAMMING TOOLS**

**1.** Produce an annotated diagram of the IDE you prefer to use to write code, explaining any features of the IDE that help you to produce your code. You may need to show examples of the IDE in use to highlight the different features.

**CODING CHALLENGES (OPTIONAL)**

The coding challenges below will let you check your skills. Part of the transition to A-level is combining skills, and also ensuring that you plan and test your work thoroughly, so think about how you can reuse components and design your code for readability and robustness.

**1.** Write an program to:

a. Ask the user to input

i. Their first name

ii. Their surname

iii. A date, in the format DD/MM/YYYY

b. The program should then output a customer ID as follows:

i. The date in the format YYYYMMDD, then the first three letters of the surname, then the first initial, then the length of their first name. All letters should be in capitals

ii. For example, John Smith, 27/05/2017 would give 20170527SMITHJ4

c. The program should validate any inputs and keep asking for inputs until the user enters correct details or types “quit” at any point

Plan your algorithm first, using a flowchart or pseudocode.

Code your algorithm, and provide evidence of both your code and the working output.

**2.** Write a program to:

a. Ask the user to input

i. The name of a product

ii. Its cost in pounds

iii. The program should keep asking for inputs until the user types “None”

b. The program should then output:

i. The name and price of the most expensive item

ii. The name and price of the least expensive item iii. The average price of the items

iv. The total cost of the items

1. Items over £50 get a 5% discount

2. VAT is added at the end at 20%

c. The program should validate any inputs

Plan your algorithm first, using a flowchart or pseudocode Code your algorithm, and provide evidence of both your code and the working output.

Create a test plan for your algorithm, including testing your validation with normal, boundary and erroneous data.