Salvatorian
Sixth Form

Salvatorian Sixth Form
GCSE to A-level Computer Science
Transition Material
Student Pack

Salvatorian
Sixth Form

Contents

Salvatorian Sixth Form

Welcome to Your A-Level Computer Science Transition Pack!

Transitioning from GCSE to A-Level Computer Science is a significant step. This pack is designed to help you bridge that gap, focusing on technical content, detailed answers, and independent research. Whether you've studied GCSE Computer Science or are new to the subject, this pack will support your A-Level preparation.

What's Inside?

1. Computer Science Theory
2. Algorithmic Thinking and Problem Solving
3. Writing Code

Each section contains practice questions to enhance your knowledge and skills, with resources for further study.

Why This Pack?

- Familiar Content : Based on the GCSE specification, it will feel familiar if you've studied GCSE Computer Science.
- Skill Building : Questions range from basic to advanced, helping you develop your thinking, writing, and problem-solving abilities.
- Challenging Questions : Tackle tough questions to prepare for A-Level complexities.

How to Use This Pack

- Explore Each Section : Start with theory, move to algorithmic thinking, and finish with coding.
- Practice and Reflect : Use the questions to practice and improve your skills.
- Utilize Resources : Follow the provided links to deepen your understanding.

This pack is a comprehensive resource to help you master key topics before starting your A-Level journey. Let's make this transition an exciting and rewarding experience!

Teach-ict login website: http://www.teach-ict.com/

Username**: ha35dy**

Password**: student**

Salvatorian Sixth Form

Computer Science Theory

Wider computing issues and integrated questions

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 you 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. Choose how you will present your ideas.

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

Salvatorian Sixth Form

Memory

1. Compare RAM and ROM

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

3. Describe the characteristics of flash memory

Storage

1. Complete the following table comparing optical, magnetic and solid state storage media

	Capacity	Speed	Portability	Durability	Reliability	Cost
Optical						
Magnetic						
Solid State						

2. Justify one use of each storage method

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
5. There have been many recent high-profile cyber-attacks across the world, including the attack on the NHS in May 2017. Some commentators have said that “we now rely too much on technology”. Write an essay explaining how far you agree with this statement and including descriptions of threats to IT systems and ways to reduce vulnerabilities.

Systems Software

1. Create a presentation comparing Windows, Linux, iOS, Android (which is based on Linux) and Unix. Discuss the features of each operating system, comparing the benefits and limitations of each. Note that you can try a basic Unix interface here: http://www.masswerk.at/jsuix/.

Introduction to Data Structures and Algorithms (DSA)

https://visualgo.net/en

Data Structures and Algorithms (DSA) are fundamental concepts in computer science that help us efficiently manage and manipulate data. Here, we'll introduce the main five data structures, explain what they are, and provide a research task to deepen your understanding.

The Main 5 Data Structures

1. Arrays
   - Definition : A collection of elements identified by index or key.
   - Characteristics : Fixed size, fast access, and can store elements of the same type.

2. Linked Lists
   - Definition : A sequence of nodes where each node contains data and a reference to the next node.
   - Characteristics : Dynamic size, efficient insertions/deletions, and sequential access.

3. Stacks
   - Definition : A collection of elements with Last In, First Out (LIFO) access.
   - Characteristics : Allows push (add) and pop (remove) operations at one end (the top).

4. Queues
   - Definition : A collection of elements with First In, First Out (FIFO) access.
   - Characteristics : Allows enqueue (add) at the rear and dequeue (remove) from the front.

5. Trees
   - Definition : A hierarchical structure with a root node and child nodes, forming a parent-child relationship.
   - Characteristics : Used to represent hierarchical data, such as file systems, and support efficient search and traversal operations.

Research Task

To gain a better understanding of these data structures, complete the following research

Salvatorian Sixth Form

tasks:

1. Definition and Usage
   - Find the precise definitions and common uses for arrays, linked lists, stacks, queues, and trees.
   - Provide real-world examples of each data structure.

2. Operations and Efficiency
   - Research the main operations (e.g., insertion, deletion, access) supported by each data structure.
   - Explain the time complexity (Big O notation) for these operations.

4. Advantages and Disadvantages
   - Compare and contrast the advantages and disadvantages of each data structure.
   - Identify scenarios where one data structure is preferred over the others.

5. Visualization
   - Find or create visual representations of each data structure to better understand their organization and operation.
   - Share your visualizations and explain how they help in understanding the data structures.

By completing this research task, you will develop a deeper understanding of the main data structures, their uses, and their implementations, setting a solid foundation for further studies in computer science.

Ethical, Legal, Cultural and Environmental Concerns

Find a recent news story on one of the following topics:

- A legal issue in computing, such as a breach of the Data Protection Act
- An ethical issue in computing, such as the development of AI
- An environmental issue in computing, such as the disposal of waste equipment
- A technical development in computer science, such as the Internet of Things

Summarise the story, explaining any technical content for a student in year 10.
Explain how the story affects you as a student of computer science.

Computational Thinking – Theory

Computational Logic and Calculations

1. Complete the truth tables for the following expressions
    a. 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			

    b. (NOT A) OR (NOT B)
        i. What single logic gate produces the same result as this expression?

A	**B**			

    c. Draw a circuit to represent each expression

Salvatorian Sixth Form

2. Calculate each of the following, showing any appropriate working you need
    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. Covert 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)

6. Check if these are valid ASCII characters. If they are, give their character equivalent. Note that the first bit is a check digit using even parity, and the remaining 7 bits are the character
    a. 1110 0010
    b. 1100 0111
    c. 0011 0110
    d. 1100 1010

Programming Tools and Standards

1. Compare the use of jpg, png and gif to store images, explaining the benefits, properties and uses of each image format

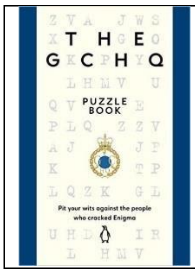
2. Produce an annotated diagram of the IDE you prefer to use to write code, explaining any features of the IDE that help you to produce your code. You may need to show examples of the IDE in use to highlight the different features

Code breaking

Decrypt the following message

WYRAC WWDEE OBORI EIOWO NUILN UEKYL CPNRD HODLO HVEMF NHRIE OYIDA NEETW T

If you like code breaking, or just really hard puzzles, try this:

	The GCHQ Puzzle Book GCHQ Michael Joseph (20 Oct. 2016) ISBN-10: 0718185544 ISBN-13: 978-0718185541	A proper work-out for the brain!

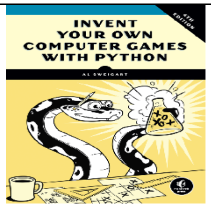
There are many more good logic, lateral and algorithmic puzzles available online if you are interested. Here is one starting point: https://en.wikibooks.org/wiki/Puzzles

Salvatorian Sixth Form

Writing Code

There are many good resources for learning to write code and to practice your coding skills.

Here are some you might like to try. Your teacher may recommend particular resources, or direct you to complete specific exercises.

	Think Python Learning with Python 3 Peter Wentworth, Jeffrey Elkner, Allen B. Downey, Chris Meyers Oct. 2012 http://openbookproject.net/thinkcs/python/english3e/index.html	One of the best free books on learning to program using python. The emphasis is on understanding why we write code and solve problems in a particular way, which is useful for A-level students. The book is well organised, with plenty of exercises in each chapter, plus a glossary of key words. Up to Ch14 is AS level, the rest of the book covers A level standard code, including the key data structures and algorithms. Note that the same resource is available for other languages, namely *Think Java* and *Think C*
	Invent with Python Albert Sweigart http://inventwithpython.com/	A nice way to start python, this site has a collection of introductory books on writing code, also all free! Each chapter has a game (or similar to make) and includes the full code, plus a step-by-step walkthrough of how to make it. It is a good exercise to read code before you write it, so making some of these games is useful.
British Informatics Olympiad	The British Informatics Olympiad https://www.olympiad.org.uk/problems.html	Lots of hard coding challenges. Like the maths challenge, only for programming! The Mayan Calendar is a good starting point.

Salvatorian Sixth Form

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 re-use 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

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

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.

Mini Project: HTML and JavaScript Basics

https://www.youtube.com/watch?v=HGTJBPNC-Gw
https://www.youtube.com/watch?v=I5kj-YsmWjM

Mini Project: HTML and JavaScript Basics

Welcome to your mini project on HTML and JavaScript! This project introduces you to the basics of web development, focusing on HTML (HyperText Markup Language) and JavaScript (JS). Whether you are new to coding or looking for a challenge, this project offers something for everyone.

Introduction to HTML

HTML is the standard language used to create web pages. It uses tags to structure and display content on the web. Here are some basic HTML tags:

- `<html>`: The root element of an HTML page.
- `<head>`: Contains meta-information about the document.
- `<title>`: Sets the title of the web page.
- `<body>`: Contains the content of the web page.
- `<h1>` to `<h6>`: Header tags, with `<h1>` being the largest and `<h6>` the smallest.
- `<p>`: Paragraph tag for text.
- `<a>`: Anchor tag for links.
- `<img>`: Tag for images.
- `<ul>` and `<li>`: Unordered list and list item tags.
- `<ol>` and `<li>`: Ordered list and list item tags.

Mini Project Tasks

Basic Tasks (For Beginners)

1. Create a Simple Web Page :
   - Use HTML tags to create a basic web page with a title, heading, paragraph, and an image.
2. Add Links and Lists :
   - Include an anchor tag that links to your favorite website and a list of your favorite hobbies.

Intermediate Tasks (For Those Ready for More)

1. Interactive Button :
   - Add a button to your web page that, when clicked, displays a message using JavaScript.
2. Change Content with JavaScript :
   - Create a function that changes the text of a paragraph when a button is clicked.

Advanced Task: Create a Calculator

Salvatorian Sixth Form

Steps to Create the Calculator

1. HTML Structure :
   - Create the basic layout with buttons for numbers 0-9, operators (+, -, *, /), and a display area.

2. CSS Styling :
   - Style the calculator using CSS to make it visually appealing.

3. JavaScript Functionality :
   - Add JavaScript to handle button clicks, perform calculations, and update the display.

This project will help you integrate HTML, CSS, and JavaScript to create a functional web application, enhancing your web development skills.

Algorithmic Thinking and Problem Solving (optional)

https://www.practicepython.org/

PracticePython.org provides over 30 exercises designed to teach specific Python concepts. The exercises range from simple tasks to complex challenges, progressively increasing in difficulty to help you continuously improve.

Key Features

1. Variety of Exercises:
   - Beginner-Friendly: Start with basic tasks suitable for newcomers.
   - Progressive Difficulty: Gradually tackle more complex problems.
   - Real-World Scenarios: Learn how Python applies to practical situations.

2. Guidance and Hints :
   - Clear Instructions : Understand what needs to be accomplished.
   - Helpful Hints : Get tips without revealing solutions.

3. Community and Discussion :
   - User Solutions : Compare your answers with others.
   - Active Discussion : Engage with other learners for support and motivation.

4. Learning Resources :
   - Tutorial Links : Access additional tutorials and resources.
   - Supplementary Material : Find extra reading and examples.

Benefits

- Hands-On Learning : Reinforce knowledge through practical application.
- Skill Development : Improve coding and problem-solving abilities.
- Flexible Learning : Progress at your own pace.
- Community Support : Get support and motivation from other learners.

Getting Started

1. Visit [PracticePython.org](https://www.practicepython.org/).
2. Choose an exercise.
3. Solve the problem and compare your solution with others.
4. Participate in discussions to deepen your understanding.

https://www.youtube.com/watch?v=XKHEtdqhLK8

Salvatorian Sixth Form

What strategy should you follow to always win at this game, or at least never lose?

Enrichment Activities to keep you busy during the Summer!

National Museum of Computing, Bletchley Park (Near Milton Keynes)

http://www.tnmoc.org/

https://www.bletchleypark.org.uk/

http://www.codesandciphers.org.uk/bletchleypark/ (virtual tour)

The National Museum of Computing and the Bletchley Park code breaking exhibition are both on the same site, although each has a separate entrance fee. Huge range of technology to explore, including Colossus, the world's first electronic computer.

Museum of Science and Industry, Manchester

http://msimanchester.org.uk/

The museum has an exhibition covering the development of computers, and they have "Baby" the world's first stored program computer. (There is an interactive talk about Baby every day.)

Science Museum, London

http://www.sciencemuseum.org.uk/

A wide range of science and technology exhibitions. In particular, the museum is currently hosting an exhibition on robotics, charting our 500 year quest to make machines human.

Centre for Computing History, Cambridge

http://www.computinghistory.org.uk/

A large collection of vintage and retro computers, with an emphasis on how computers have developed over time and the social context and impact of technological change.