

# What is Computer Science? A Case Study for Intro to CS Students

---

CASE STUDIES IN COMPUTER SCIENCE | BRENDAN SHEA, PHD

In our increasingly digital world, computer science has become a fundamental field that shapes nearly every aspect of our lives. But what exactly is computer science, and why is it so important? This case study will explore the essence of computer science, its impact on our daily lives, and its potential to shape our future.

## The Ubiquity of Algorithms

At the heart of computer science lies the concept of **algorithms**. An **algorithm** is a step-by-step procedure for solving a problem or accomplishing a task. In the digital realm, algorithms are the invisible force guiding our experiences and making decisions behind the scenes.

Consider your typical morning routine. Your smartphone's alarm wakes you up, using an algorithm to determine the optimal time based on your sleep cycles. As you check your social media over breakfast, algorithms curate your feed, deciding which posts to show you based on your past interactions and preferences. When you navigate to work, GPS algorithms calculate the fastest route, taking into account real-time traffic data.

These examples illustrate how algorithms have become deeply integrated into our daily lives:

1. Whether you're shopping online or streaming music, algorithms analyze your behavior and preferences to suggest products or content you might enjoy.
2. Algorithms ensure the security and efficiency of our financial systems, from credit card fraud detection to high-frequency trading.
3. Algorithms assist in diagnosing diseases, analyzing medical images, and even predicting patient outcomes.
4. Email spam filters, autocorrect, and language translation services all rely on sophisticated algorithms to function.
5. From thermostats that learn your temperature preferences to voice assistants that interpret your commands, algorithms make our homes "smarter."

The pervasiveness of algorithms underscores the importance of understanding computer science in today's world. It's not just about programming; it's about developing the skills to understand and interact with the digital systems that increasingly mediate our experiences.

## Defining Computer Science

So, what exactly is computer science? At its core, computer science is the study of computation, information processing, and the design of computer systems. It encompasses a wide range of topics, from the theoretical foundations of computation to practical applications in software engineering and artificial intelligence.

Computer science is often described as the "study of algorithms" because of its fundamental focus on developing systematic procedures for solving problems. This description captures a key aspect of the field, but it's important to understand that computer science is much broader than just writing code.

### Key Components of Computer Science:

Component	Description
-----------	-------------

---

<b>Algorithm Design</b>	Creating step-by-step procedures to solve problems efficiently
<b>Data Structures</b>	Organizing and storing data for effective access and modification
<b>Programming Languages</b>	Designing and using languages to instruct computers
<b>Computer Architecture</b>	Understanding and designing the hardware that executes algorithms
<b>Artificial Intelligence</b>	Developing systems that can learn and make decisions
<b>Networking</b>	Designing systems for computers to communicate and share resources
<b>Cybersecurity</b>	Protecting digital systems and data from unauthorized access

One of the most powerful concepts in computer science is **abstraction**. **Abstraction** involves simplifying complex systems by focusing on the essential details while ignoring irrelevant ones. This allows computer scientists to tackle incredibly complex problems by breaking them down into manageable parts and building solutions layer by layer.

## The Art of Problem-Solving in Computer Science

Computer scientists approach problem-solving in diverse ways, employing various methodologies and techniques. Here are some key approaches:

1. **Theoretical Analysis.** Computer scientists study the mathematical properties of algorithms, analyzing their **time complexity** (how long they take to run) and **space complexity** (how much memory they use). This theoretical work provides insights into the fundamental limits of computation and guides the development of practical algorithms.
2. **Algorithm Design Paradigms.** Computer scientists use various design strategies to create efficient solutions to complex problems. For example, the **divide-and-conquer** paradigm involves breaking a problem into smaller subproblems, solving these subproblems, and then combining the results. This approach is used in many efficient sorting algorithms.
3. **Empirical Testing.** While theoretical analysis is important, the performance of algorithms in practice can sometimes differ from theoretical predictions. Computer scientists implement algorithms and test them with real-world data to evaluate their practical performance.
4. **Machine Learning Approaches.** With the rise of artificial intelligence, many computer scientists now focus on creating algorithms that can learn from data. These algorithms can automatically improve their performance with experience, leading to systems that can adapt to new situations and handle complex, poorly-defined problems.
5. **Optimization.** This involves developing algorithms to find the best solution among many possibilities. Optimization algorithms are used in a wide range of applications, from logistics and supply chain management to financial portfolio allocation.

The diversity of these approaches reflects the breadth and depth of computer science as a field. By approaching problems from multiple angles, computer scientists continue to push the boundaries of what's possible in computation, driving innovation across numerous industries and scientific disciplines.

## Future Developments in Computer Science

As we look to the future of computer science, several exciting developments are on the horizon:

1. **Quantum Computing** promises to solve certain problems exponentially faster than classical computers. Quantum computers operate on fundamentally different principles, exploiting properties like

superposition and entanglement. As quantum hardware improves, we can expect new quantum algorithms that could tackle previously intractable problems in fields like cryptography and drug discovery.

2. As AI is increasingly used to make important decisions, there's a growing need for algorithms that can explain their decision-making processes. **Explainable AI (XAI)** aims to create AI systems that are not just accurate, but also interpretable by humans.
3. We're likely to see more AI processing happening on local devices (**edge computing**) rather than in the cloud. This could lead to faster, more private AI applications, as sensitive data wouldn't need to be sent to remote servers for processing.
4. **Neuromorphic computing** involves designing computer architectures inspired by the human brain. Unlike traditional architectures, neuromorphic systems could potentially process information in a way that's much more energy-efficient and better suited for AI tasks.
5. As AI becomes more pervasive, we're likely to see the development of new **ethical frameworks** dedicated to ensuring AI systems are fair, transparent, and beneficial to society.

## The Value of Computational Thinking

While these advanced topics might seem relevant only to computer scientists, the fundamental skills of computer science - often referred to as **computational thinking** - are valuable for everyone, regardless of their career path.

**Computational thinking** is a problem-solving approach that draws on concepts fundamental to computer science. It involves several key components:

- **Decomposition.** Breaking complex problems into smaller, more manageable parts.
- **Pattern Recognition.** Identifying similarities or common differences that can help in problem-solving.
- **Abstraction** Focusing on the essential information while ignoring irrelevant details.
- **Algorithmic Thinking.** Developing a step-by-step solution to the problem.

These skills are valuable far beyond the realm of computer programming. Here's why computational thinking is important for everyone:

1. Computational thinking provides a structured approach to tackling complex problems in any field. Whether you're a business strategist developing a market entry plan or a city planner designing traffic flow, the ability to break down problems and develop systematic solutions is invaluable.
2. In our increasingly data-driven world, the ability to collect, analyze, and draw insights from data is becoming crucial in many professions. Computational thinking provides a foundation for understanding how to approach data analysis, even for those who aren't data scientists.
3. By emphasizing systematic thinking and cause-and-effect relationships, computational thinking enhances our ability to think logically and make reasoned decisions. This is beneficial in both personal and professional contexts, from making financial decisions to developing business strategies.
4. As technology continues to evolve, the ability to think computationally becomes increasingly valuable. It provides a framework for understanding and interacting with digital systems, even as they become more complex.
5. While computational thinking is often associated with logical processes, it can also enhance creativity. By providing tools to break down and analyze problems, it can lead to innovative solutions that might not be immediately apparent.

As you can see, computer science is far more than just programming or building computers. It's a diverse and dynamic field that touches virtually every aspect of our modern world. From the algorithms that shape our daily digital experiences to the cutting-edge developments in AI and quantum computing, computer science continues to push the boundaries of what's possible.

Moreover, the problem-solving approaches central to computer science - encapsulated in computational thinking - offer valuable skills for everyone in our increasingly digital and data-driven world. Whether you're a future computer scientist or simply a citizen of the digital age, understanding the basics of computer science and cultivating computational thinking skills can help you navigate and thrive in the technology-rich landscape of the 21st century.

## Discussion Questions: What is Computer Science?

1. Think about an app or website you use regularly. Can you identify and describe at least three ways that algorithms might be working behind the scenes to enhance your user experience?
2. How might you apply computational thinking to organize a large school event, like a sports tournament? Break down your approach using the four components of computational thinking.
3. Imagine you're designing a simple AI to play tic-tac-toe. What kind of data would you need to collect to train this AI, and how might you go about collecting it?
4. Consider the concept of edge computing discussed in the case study. How might this technology improve the functioning of smart home devices or autonomous vehicles?
5. The case study mentions that quantum computing could solve certain problems exponentially faster than classical computers. Research and explain the concept of "quantum supremacy". What might be some potential applications of this technology?
6. Compare and contrast the divide-and-conquer algorithm design paradigm with dynamic programming. Can you think of a real-world problem that might be suited to each approach?
7. Explore the concept of P vs NP problems in computer science. Why is this considered one of the most important open questions in the field? What would be the implications if P were proven to equal NP?
8. The case study mentions neuromorphic computing as a future development. How does this differ from traditional von Neumann architecture, and what advantages might it offer for certain types of computational tasks?
9. As AI systems become more prevalent in making important decisions (like in healthcare or criminal justice), what ethical considerations should be taken into account? How can we ensure these systems are fair and unbiased?
10. Discuss the potential societal impacts of widespread quantum computing. How might it affect fields like cryptography and online security? What might be some positive and negative consequences?
11. Consider the increasing use of AI in creative fields like art and music. Does AI-generated art challenge our understanding of creativity and authorship? Should AI-generated works be eligible for copyright protection?
12. The case study suggests that computational thinking is valuable for everyone. Do you agree? Should computational thinking be a required subject in high schools or colleges? What might be the benefits and drawbacks of such a requirement?