Introduction to Computer Science

Publication Version: 2024.12.20

Prepared and Published by

Computer Science (CS) is a fascinating and vast field that underpins much of modern technology, from the software that runs on your smartphone to the algorithms that manage global financial systems. This introduction will cover the foundational aspects of CS, providing a broad overview for beginners or those curious about what computer science entails, including a look at historical systems like Paradox, a notable Relational Database Management System (RDBMS).

What is Computer Science?

At its core, computer science is the study of computation, information, and automation. It involves designing and understanding algorithms, developing and analyzing software systems, and exploring the theoretical foundations of information processing. Unlike computer engineering, which focuses more on hardware, CS is primarily concerned with software and the abstract principles that govern it.

This discipline also encompasses a broad range of subfields such as artificial intelligence, where algorithms are designed to mimic human cognition; data science, which deals with extracting insights from large datasets; and software engineering, which focuses on the systematic approach to software development. Through these areas, computer science not only solves practical problems but also pushes the boundaries of what machines can achieve, influencing nearly every aspect of modern life from healthcare to entertainment.

Historical Context

- **Origins:** The roots of computer science trace back to the 19th century with Charles Babbage's Analytical Engine, an early mechanical general-purpose computer. Ada Lovelace's notes on this machine are often regarded as the first computer program.

- **Development:** The field truly took off with the advent of electronic computers in the mid-20th century. Pioneers like Alan Turing, whose work on the Turing machine laid theoretical groundwork, and later, the development of programming languages like FORTRAN (1957) and C (1972).

- **Expansion:** The late 20th and early 21st centuries saw an explosion in computing power and the internet, leading to new areas like network theory, cybersecurity, and the advent of personal computing, which democratized access to computational resources.

Core Areas of Computer Science

1 **Algorithms and Data Structures:**

   a. **Algorithms:** These are step-by-step procedures for solving problems. Understanding algorithms is crucial as they determine how efficiently a task can be performed.

   b. **Data Structures:** Ways to organize and store data so that it can be accessed and modified efficiently. Examples include arrays, linked lists, trees, and graphs.

2 **Programming Languages:**

   a. Languages like Python, Java, C++, and JavaScript are tools used to write software. Each has its strengths; for instance, Python is known for simplicity and readability, making it ideal for beginners.

3 **Theory of Computation:**

   a. This branch deals with whether problems can be solved by a computer and how. Concepts like computability, complexity theory, and formal languages are explored here.

4 **Software Engineering:**

   a. Focuses on the methodologies for creating software, including design, development, maintenance, and testing. It emphasizes teamwork, project management, and software quality.

5 **Computer Architecture:**

   a. While primarily hardware-focused, understanding how computers are architected helps in designing more efficient software. Topics include memory management, CPU design, and I/O systems.

6 **Networking and Security:**

   a. With the rise of the internet, understanding network protocols, security measures, and data protection has become crucial. This includes cryptography, network security, and the study of cyber threats.

7 **Artificial Intelligence (AI) and Machine Learning (ML):**

   a. AI aims to create machines that work and react like humans. ML, a subset of AI, involves developing algorithms that can learn from data to make predictions or decisions without explicit programming.

Modern Applications

- **Big Data:** Managing and analyzing vast amounts of data to derive insights.
- **Internet of Things (IoT):** Embedding computational capabilities into everyday objects.
- **Cloud Computing:** Delivering computing services over the internet, offering scalable, on-demand computing resources.

Database Development

Databases are crucial components of modern software applications, providing structured ways to store, retrieve, update, and manage data efficiently. Database development involves designing, implementing, and maintaining databases to support various applications.

Key Concepts in Database Development:

- **Database Models:**
  - **Relational Databases:** Use tables to store data, with relationships defined by keys. SQL (Structured Query Language) is commonly used to manage these databases.
  - **NoSQL Databases:** Handle unstructured data and are scalable, often used for big data applications, including document-oriented databases.
- **Normalization:** A technique to minimize redundancy and dependency by organizing data into tables in such a way that addition, deletion, and modification of data can be made more efficient.
- **Database Design:**
  - **Entity-Relationship Diagrams (ERD):** Used to visualize the database structure, showing entities, their attributes, and relationships.
  - **Schema:** The structure of the database described in a formal language supported by the database management system (DBMS).
- **Query Languages:**
  - **SQL:** Standard language for managing relational databases, with commands for CRUD operations (Create, Read, Update, Delete).
- **Database Management Systems (DBMS):**
  - Software like MySQL, PostgreSQL, Oracle, and MongoDB, which manage and control databases, and Paradox (RDBMS) for desktop database systems.

Paradox and ObjectPAL

With a rich history in the desktop database market, Paradox, sitting at version 11, is a powerful relational database management system (RDBMS) that runs on Microsoft Windows 10 and 11. It offers robust features for data management, query handling, and application development. Paradox is particularly beneficial for small businesses and departments within larger companies due to its user-friendly interface and powerful data handling capabilities. It allows businesses to efficiently manage large volumes of data, generate comprehensive reports, and customize forms to meet specific operational needs. The ability to automate repetitive tasks reduces manual labor, increases productivity, and minimizes errors.

ObjectPAL (Object-oriented Paradox Application Language), the integrated programming language of Paradox, allows developers to create complex desktop applications and automate tasks within the database environment. By leveraging ObjectPAL, users can customize forms and reports, and write scripts to validate data, enhancing functionality and user experience. ObjectPAL offers a rich set of features, including event-driven programming, robust error handling, and support for complex data manipulation. This makes it an ideal tool for creating sophisticated business applications, performing data analysis, and integrating with other software systems. Additionally, ObjectPAL's intuitive syntax and comprehensive documentation make it accessible for both novice and experienced developers, enabling them to quickly develop and deploy powerful database solutions. With ObjectPAL, the possibilities for automation and customization are virtually limitless, providing a flexible and scalable platform for businesses of all sizes.

More About Paradox:
- **Background:** Introduced by Ansa Software in 1985, Paradox is one of the early PC-based relational database management systems, later acquired by Borland Corporation in 1987, then by Corel Corporation in 1996.

- **Features:** It offers a user-friendly interface for managing databases without deep programming knowledge. Paradox is known for its integration with Corel's other products, like Quattro Pro, providing seamless data handling across different applications.

- **Usage:** Initially popular with small to medium businesses for tasks like inventory management, customer databases, and simple accounting solutions. It supports both a graphical user interface and SQL-like query language for data manipulation.

More About ObjectPAL (Object-oriented Paradox Application Language):

- **Purpose:** ObjectPAL is the proprietary programming language of Paradox, which provides an object-oriented approach to database application development. It is designed to make it easier to create complex applications by leveraging object-oriented programming principles.

- **Features:**
  - **Event-Driven Programming:** ObjectPAL allows developers to write scripts that respond to user actions or system events, making applications interactive.

  - **Integration with Database:** It's tightly integrated with Paradox's database operations, allowing for direct manipulation of database objects through code.

  - **User Interface Development:** Developers could create custom forms, reports, and menus directly within the Paradox environment, enhancing the user experience.

- **Advantages:**
  - **Ease of Use:** For developers familiar with Paradox's environment, ObjectPAL offers a straightforward way to extend functionality without needing to switch to a different programming environment.

  - **Rapid Development:** The integration with the DBMS means that developers can rapidly prototype applications, testing them directly against live data.

- **Limitations:**
  - **Proprietary:** Being tied to Paradox means that knowledge and scripts written in ObjectPAL are not easily transferable to other database systems.

  - **Adoption:** As the industry continues to move towards more standardized languages and platforms, Paradox and ObjectPAL has seen reduced usage in favor of systems like SQL Server, Oracle, or open-source solutions like PostgreSQL.

Current Relevance: While Paradox and ObjectPAL are less prevalent today, and no product updates have been released in many years, understanding these systems provides insight into the evolution of database management and application development. They highlight the shift from proprietary, closed systems to more open, interoperable solutions in database technology. However, for small businesses, Paradox and ObjectPAL can still hold relevance. These tools offer a straightforward approach to database management and custom application development, which can be particularly beneficial for businesses with limited resources or those looking for cost-effective, legacy solutions. This can be especially true for small enterprises that have existing systems built on these platforms or require simple, yet functional, database solutions without the complexity of modern, large-scale database systems.

Conclusion

Computer science is not just about coding; it's the study of algorithms, computation, and information systems, focused on solving problems and advancing technology through computational methods. As an expansive field, it offers a wide range of opportunities for anyone interested in learning more.