Answers are in blue.

# For Exercises 1–10, match the following people with their accomplishments.



**Computer Science Illuminated, Seventh Edition**

Nell Dale, PhD; John Lewis, PhD

**CHAPTER 1**

EXERCISES AND ANSWERS

1. **Leibniz**
2. **Pascal**
3. **Babbage**
4. **Lovelace**
5. **Hollerith**
6. **Byron**
7. **Turing**
8. **Jacquard**
   1. What French mathematician built and sold the first gear- driven mechanical machine that did addition and subtraction? B
   2. Who built the first mechanical machine that did addition, subtraction, multiplication, and division?

## A

* 1. Who designed the first mechanical machine that included memory?

## C

* 1. Who is considered the first programmer? D
  2. Who proposed that a punched card be used for counting the census?

## E

* 1. Who edited Babbage’s work? D
  2. Who was Ada Lovelace’s father? F
  3. Who would have been mentioned in the book *The Code Breakers*?

## G

* 1. Who developed the concept of punched holes used in weav- ing cloth?

## H

* 1. Who is associated with IBM? E

# For Exercises 11–23, match the hardware listed to the appropriate generation.

* + 1. **First**
    2. **Second**
    3. **Third**
    4. **Fourth**
    5. **Fifth**
  1. Circuit boards C
  2. Transistor

## B

* 1. Magnetic core memory 磁芯存储器

B

* 1. Card input/output A
  2. Parallel computing

并行计算

D

* 1. Magnetic drum A
  2. Magnetic tape drives A
  3. Integrated circuits C
  4. Personal computer D
  5. Vacuum tube A
  6. Large-scale integration D
  7. Magnetic disc B
  8. Networking D

# For Exercises 24–38, match the software or software concepts listed to the appropriate generation.

* + 1. **First**
    2. **Second**
    3. **Third**
    4. **Fourth**
    5. **Fifth**
  1. AssemblerA

## FORTRAN

B

* 1. Operating systems C
  2. Structured programming 结构化程序设计

D

* 1. Time sharing C
  2. HTML (for the Web) E
  3. Loaders装入器B
  4. Spreadsheets D
  5. Word processors D
  6. Lisp B

## PC-DOS D

* 1. Loaders/linkers bundled into operating system C
  2. Java E
  3. SPSS C
  4. C D

# Exercises 39–59 are short answer.

* 1. What do we mean by the statement that “the 1980s and 1990s must be characterized by the changing profile of the user”?

The original user was the programmer who had a problem to solve. By the 1970s, application programs were being written such that nonprogrammers could use them to solve prob- lems. With the advent of the personal computer, many peo- ple began using the computer for personal correspondence, personal accounts, and games.

* 1. Why was Mosaic important?

Mosaic was the first graphics-capable browser. It ushered in the world of user-friendly browsers such as Netscape, Inter- net Explorer, and Firefox.

* 1. Discuss the browser wars.

Netscape Navigator and Internet Explorer were the big com- peting user-friendly browsers. Microsoft’s IE was bundled with the Windows operating system, giving IE the edge, which led to a monopoly lawsuit. However, the damage was done and IE was the winner.

* 1. Describe how the Web changed after 2002.

The advent of social networking sites allowed users to pro- vide content to Internet sites. Blogging, both personal and professional, also changed the character of Internet content. This change is sometimes referred to as *Web 2.0*.

* 1. Of the predictions listed, which do you consider the biggest error in judgment?

This is a personal choice; no answer needed.

* 1. Name the four areas in which the practitioner must be skilled. Algorithmic thinking, representation (of data storage), pro- gramming, and design.
  2. Distinguish between computing as a tool and computing as a discipline.

Computing as a tool refers to the use of computing to solve problems in a person’s professional or personal life. Comput- ing as a discipline refers to the study of the body of knowl- edge that makes up computer science and/or computer engineering.

* 1. Is computing a mathematical discipline, a scientific disci- pline, or an engineering discipline? Explain.

Computing is not a mathematical discipline nor a scien- tific discipline nor an engineering discipline; it has roots in all three disciplines. Boolean algebra, logic, and numerical analysis contribute greatly to the foundations of computing. Simulation and model building from the scientific discipline contribute to the foundations of computing. The techniques from engineering of building large systems contribute to the foundations of computing.

* 1. Distinguish between systems areas and applications areas in computing as a discipline.

The systems areas of computing as a discipline relate to the understanding and building of computer tools: algorithms and data structures, programming languages, (computer) architecture, operating systems, software methodology and engineering, and human–computer communication. The applications areas in computing relate to the computer’s use as a tool: numerical computation, databases and informa- tional retrieval, artificial intelligence and robotics, graphics, organizational informatics, and bioinformatics.

* 1. Define the word *abstraction* and relate it to the drawing in Figure 1.2.

An abstraction is a mental model that removes complex details. An abstraction lets the viewer see only those details that are relevant to the user’s view. An engine is a mental model of a car from the builder or mechanic’s view. The cockpit is the mental model for an ordinary driver, who does not have to know the details of the engine in order to drive the car.

* 1. What is cloud computing?

The use of computer resources such as servers on the Inter- net instead of relying on devices at your physical location.

* 1. Define the word *protocol* and explain how it is used in computing.

A protocol is a code prescribing strict adherence to correct etiquette and procedure (as in a diplomatic exchange). Com- puting terminology has borrowed the word to describe the correct etiquette for computers to use when communicating with one another.

* 1. Distinguish between machine language and assembly language.

Machine language is the language that is built into the electrical circuitry of a computer. Assembly language is

a language made up of mnemonic codes that represent machine-language instructions. Programs written in assem- bly language are translated into machine-language pro- grams by a computer program called an *assembler*.

* 1. Distinguish between assembly language and high-level languages.

Whereas assembly language is a language made up of mnemonic codes that represent machine-language instruc- tions, high-level languages use English-like statements to represent groups of assembly-language statements or machine-language statements. There is a one-to-one cor- respondence between statements in an assembly language and the statements they represent in machine language. There is a one-to-many correspondence between high- level statements and the corresponding machine-language statements.

* 1. FORTRAN and COBOL were two high-level languages defined during the second generation of computer software. Compare and contrast these languages in terms of their his- tory and their purpose.

FORTRAN was written at IBM for use in solving scientific and engineering problems. It was not formally designed and has grown greatly over the years. COBOL was designed by a team for business applications and has been relatively stable.

* 1. Distinguish between an assembler and a compiler.

An assembler translates assembly-language programs into machine code. A compiler translates programs in a high- level language into either assembly-language programs or machine-language programs.

* 1. Distinguish between a systems programmer and an applica- tions programmer.

A systems programmer writes programs that are tools to help others write programs. An applications programmer writes programs to solve specific problems.

* 1. What was the rationale behind the development of operat- ing systems?

The human operator was too slow. Computers were idle while the human prepared the next program to be run. The computer has the speed to organize itself.

* 1. What constitutes systems software?

Utility programs such as loaders and linkers, operating sys- tems, and language translators are systems software.

* 1. What do the following pieces of software do?
     1. Loader

A loader puts a program’s instructions into memory, where they can be executed.

* + 1. Linker

A linker is a program that puts pieces of a large program together so that it can be put into memory, where it can be executed.

* + 1. Editor

An editor is a word processing program that allows the user to enter and edit text.

* 1. How was the program SPSS different from the programs that came before it?

SPSS was the first application program written so that the nonprogrammer user could enter data and specify the pro- cessing of the data.