Fundamentals of Java: AP* Computer Science Essentials, 4th Edition

Chapter 1: Background

A Guide to this Instructor's Manual:
We have designed this Instructor's Manual to supplement and enhance your teaching experience through classroom activities and a cohesive chapter summary.

This document is organized chronologically, using the same headings in plum that you see in the textbook. Under each heading you will find (in order): Chapter Objectives, Teacher Materials, a Prepare section that focuses students' attention on the objectives in the chapter, Instructor Notes and Teacher Tips that summarize the section, Figures and Boxes found in the section, Classroom Activities, Projects to Assign, Key Terms, and Assess. Pay special attention to teaching tips, and activities geared towards quizzing your students, enhancing their critical thinking skills, and encouraging experimentation within the software.

In addition to this Instructor's Manual, our Instructor Resources CD also contains PowerPoint presentations, Test Banks, and other supplements to aid in your teaching experience.

For your students:
Our latest online feature, CourseCasts, is a library of weekly podcasts designed to keep your students up to date with the latest in technology news. Direct your students to http://coursecasts.course.com, where they can download the most recent CourseCast onto their mp3 players. Ken Baldauf, host of CourseCasts, is a faculty member of the Florida State University Computer Science Department where he is responsible for teaching technology classes to thousands of FSU students each year. Ken is an expert in the latest technology and sorts through and aggregates the most pertinent news and information for CourseCasts so your students can spend their time enjoying technology, rather than trying to figure it out. Open or close your lecture with a discussion based on the latest CourseCast.

Table of Contents

Chapter Objectives

Students will have mastered the material in Chapter 1 when they can:

- Give a brief history of computers.
- Describe how hardware and software make up computer architecture.
- Explain the binary representation of data and programs in computers.
- Discuss the evolution of programming languages.
- Describe the software development process.
- Discuss the fundamental concepts of object-oriented programming.

Teacher Materials

- Learner text.
- PowerPoint presentation from the **PowerPoint Presentations** drop-down menu on the *Instructor Resources* CD.
- Solutions to review questions and projects from the **Solutions to Exercises** drop-down menu on the *Instructor Resources* CD.
- ExamView® test questions from the **Test Bank & Test Engine** drop-down menu on the *Instructor Resources* CD.

Prepare

- Set up a projection system and show the PowerPoint presentation for the chapter, if desired.
- Give necessary instructions to students about lab guidelines.
- Prepare questions from ExamView.

Technical Notes

- No software is used in this lesson.

Instructor Notes and Teacher Tips

- Review Objectives and ask students what they know about the history of computers.
- Review the vocabulary list at the beginning of the chapter, and ask students to identify any terms they are already familiar with.

3: 1.1 History of Computers

- Review the changes that each decade of computing has brought.
- Define ubiquitous computing, and ask students to identify the personal devices and public computers with which they interact on a daily basis.

- Discuss why it is important to understand the history of computers, define computing terms, and discuss the development of programming languages before learning a programming language such as Java.

FIGURES

- Figure 1.1: An interconnected world of computers

CLASSROOM ACTIVITIES

1. Quick Quizzes:
   1) True or False? The mean time between failures of the ENIAC was less than an hour.
   Answer: True.
   2) What kind of computer supports several users simultaneously while giving each person the illusion that he or she "owns" the computer?
   A. PC
   B. hand-held calculator
   C. time-sharing computer
   Answer: C
2. Class Discussion:
   1) In the 1950s it was thought that the world needed no more than 10 business computers. Discuss the differences between 1950s society and that of today.

5: 1.2 Computer Hardware and Software

- The difference between hardware and software is an important aspect of this chapter.
- Students should have a broad understanding of hardware components and how they fit together. Use Figure 1-2 to discuss a computer's six major subsystems, and see if students can come up with other examples of devices in the auxiliary I/O, auxiliary story, and user interface categories.
- Discuss the role of the central processing unit (CPU) in the computer's operations. Make sure students understand Moore's Law, the transistor, and how the CPU interacts with RAM to run the computer.
- Students should also become familiar with system software, such as the computer's operating system and the services it provides.
- Ask students about their experiences with application software, including packages such as Microsoft Office, and online software suites such as Google docs.
- It is important for them to understand that programmers modify software to add new services to a computer and rarely modify hardware.
- In Exercise 1.2, students answer questions that test their understanding of the concepts of hardware and software. Review these questions and students' answers in class to make sure they understand these concepts.

FIGURES

- Table 1-1: Some commonly used quantities of information storage
- Figure 1-2: A computer's six major subsystems

CLASSROOM ACTIVITIES

1. Quick Quizzes:
   1) True or False? Secondary memory typically has a much greater storage capacity than primary memory.
      Answer: True.
   2) True or False? Word processors translate programs to executable form.
      Answer: False. Compilers translate programs to machine language; word processors are application software.
   3) A _____ consists of 8 adjacent bits.
      Answer: Byte
   4) The _____ does the work of the computer.
      Answer: CPU (central processing unit)
   5) Spreadsheets are examples of _____ software.
      Answer: Application
2. Lab Activity:
   1) Have students go into the computer lab or view pictures of computers, and identify as many examples of each of the six major subsystems as possible.

10: 1.3 Binary Representation of Information and Computer Memory

- Students should understand how computers store information. Understanding how integers, floating-point numbers, characters, strings, images, and sound are digitized (and how memory in general works) is an essential aspect of computer fluency.
- It is helpful for students to run through a few examples of the binary, octal, and hexadecimal number systems and look at bit patterns for characters.
- Ensure students know how complex numbers can be represented, including floating-point numbers and mantissal/exponent notation.
- ASCII and Unicode are two ways to represent characters when a computer processes letters, digits and symbols. Java uses Unicode.
- A general discussion of the binary representation of images, video, and sound can generate high interest among students. Make sure students understand that the storage methods compress the data needed to make the media playable, storable, and transferable.
- In Exercise 1.3, students answer questions that test their understanding of the concepts learned in this section, including how numbers are stored and translated. Review these questions and students' answers in class to make sure they understand these concepts.

BOXES

- Computer Ethics: The ACM Code of Ethics. The Association for Computing Machinery (ACM) is the flagship organization for computing professionals. The ACM supports publications of research results and new trends in computer science, sponsors conferences and professional meetings, and provides standards for computer scientists as professionals. The standards concerning the conduct and professional responsibility of computer scientists have been published in the ACM Code of Ethics. The code is intended as a basis for ethical

decision making and for judging the merits of complaints about violations of professional ethical standards.

The code lists several general moral imperatives for computer professionals:

- Contribute to society and human well-being.
- Avoid harm to others.
- Be honest and trustworthy.
- Be fair and take action not to discriminate.
- Honor property rights, including copyrights and patents.
- Give proper credit for intellectual property.
- Respect the privacy of others.
- Honor confidentiality.

The code also lists several more specific professional responsibilities:

- Strive to achieve the highest quality, effectiveness, and dignity in both the process and products of professional work.
- Acquire and maintain professional competence.
- Know and respect existing laws pertaining to professional work.
- Accept and provide appropriate professional review.
- Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.
- Honor contracts, agreements, and assigned responsibilities.
- Improve public understanding of computing and its consequences.
- Access computing and communication resources only when authorized to do so.

In addition to these principles, the code offers a set of guidelines that provide professionals with explanations of various issues contained in the principles. The complete text of the ACM Code of Ethics is available at the ACM's Web site, http://www.acm.org.

FIGURES

- Table 1-2: Some base 10 numbers and their base 2 equivalents
- Table 1-3: Some characters and their corresponding ASCII bit patterns
- Figure 1-3: A sound waveform
- Figure 1-4a: Sampling a waveform
- Figure 1-4b: Regenerating the sound from the samples
- Figure 1-5: A 32 MB RAM

CLASSROOM ACTIVITIES

1. Quick Quizzes:
   1) Computers use a base 2, or _____ notation.
      Answer: binary
   2) Which of the following generally requires more bits to represent?
      A. integer
      B. character
      C. image
      D. sound
      Answer: D
   3) 378.89 is a(n) _____ number.
      Answer: floating-point

4) Sound information is _____.
Answer: Analog
5) Java uses a scheme called _____, in which each character is represented by a pattern of 16 bits.
Answer: Unicode

2. Lab Activity:
1) Do an Internet search to find a site that translates words into binary code. Translate your own name into binary code. If available, translate it into hex and octal.

19: 1.4 Programming Languages

- Emphasize the distinction between expressing ideas in a form that people can understand and expressing ideas in a form that machines can represent and manipulate.
- Call attention to the fact that the first computers were programmed in languages that were very close in form to the languages of the machines and that programming languages have evolved to better approximate the languages that people use to express their ideas.
- Make sure students understand the three generations of programming languages, and where Java fits in.
- In Exercise 1.4, students answer questions that test their understanding of the concepts learned in this section, including the development of programming languages and the different generations. Review these questions and students' answers in class to make sure they understand these concepts.

CLASSROOM ACTIVITIES

1. Quick Quizzes:
1) Which of the following is an example of a high-level language?
A. assembly language C. Java
B. machine language
Answer: C
2) A(n) _____ translates a program in a high-level language to machine code.
Answer: compiler
3) Java is a _____ generation programming language.
Answer: third

2. Class Discussion:
1) What difficulties with coding and interpreting first and second generation programming languages led to the development of the third generation?

20: 1.5 The Software Development Process

- Running through the software development process with students is helpful. These phases should be kept in mind throughout the entire course. Even with the most simplistic programs, developing the habits of this cycle will pay off for students.
- Students may want to talk to someone who is working in the field of computer programming. Either having a guest speaker come to your classroom or assigning students to interview a

programmer would be appropriate. You can also see if you can find video of an interview on You Tube or another video-sharing site.

- Use Figures 1-7 and 1-8 to reinforce the costs for each stage of the SDLC.
- In Exercise 1.5, students answer questions that test their understanding of the concepts learned in this section, including the different phases of software development. Review these questions and students' answers in class to make sure they understand these concepts.

FIGURES

- Figure 1-6: The waterfall model of the software development life cycle
- Figure 1-7: Relative costs of repairing mistakes when found in different phases
- Figure 1-8: Percentage of total cost incurred in each phase of the development process

CLASSROOM ACTIVITIES

1. Quick Quizzes:
   1) Which of the following is the most costly part of the software life cycle?
      A. design
      B. coding
      C. testing
      D. maintenance
      Answer: D
   2) _____ is the part of the software development process wherein the programmers write the program code.
      Answer: Implementation
2. Critical Thinking:
   1) For every phase, list 1-3 guidelines to help prepare for the next phase and to ensure that the cycle goes smoothly. For example, at the first phase, customer request, asking questions such as the number of simultaneous users, type of interface needed, and system requirements will ensure that you develop the right kind of program for the client.
   2) How might a team of programmers work using the waterfall method of the SDLC to create a product, as opposed to a single programmer? What are the benefits of working as a team? What could be potential obstacles?

23: 1.6 Basic Concepts of Object-Oriented Programming

- Review the difference between object-oriented programming and procedural programming.
- Have students discuss the concepts of classes, methods, encapsulation, information hiding, polymorphism, and inheritance.
- Use Table 1-4 to compare an expedition to object-oriented programming.
- In Exercise 1.6, students answer questions that test their understanding of the concepts learned in this section, including the different terms related to object-oriented programming. Review these questions and students' answers in class to make sure they understand these concepts.

BOXES

- Computer Ethics: Copyright, Intellectual Property, and Digital Information. For hundreds of years, copyright law has regulated the use of intellectual property. At stake are the rights of

authors and publishers to a return on their investment in works of the intellect, which include printed matter (books, articles, etc.), recorded music, film, and video. More recently, copyright law has been extended to include software and other forms of digital information. For example, copyright law protects the software used with this book. This prohibits the purchaser from reproducing the software for sale or free distribution to others. If the software is stolen or "pirated" in this way, the perpetrator can be prosecuted and punished by law. However, copyright law also allows for "fair use"—-the purchaser may make backup copies of the software for personal use. When the purchaser sells the software to another user, the seller thereby relinquishes the right to use it, and the new purchaser acquires this right.

When governments design copyright legislation, they try to balance the rights of authors and publishers to a return on their work against the rights of the public to fair use. In the case of printed matter and other works that have a physical embodiment, the meaning of fair use is usually clear. Without fair use, borrowing a book from a library or playing a CD at a high school dance would be unlawful.

With the rapid rise of digital information and its easy transmission on networks, different interest groups—-authors, publishers, users, and computer professionals—-are beginning to question the traditional balance of ownership rights and fair use. For example, is browsing a copyrighted manuscript on a network service an instance of fair use? Or does it involve a reproduction of the manuscript that violates the rights of the author or publisher? Is the manuscript a physical piece of intellectual property when browsed or just a temporary pattern of bits in a computer's memory? When you listen to an audio clip on a network, are you violating copyright, or only when you download the clip to your hard drive? Users and technical experts tend to favor free access to any information placed on a network. Publishers and, to a lesser extent, authors tend to worry that their work, when placed on a network, will be resold for profit.

Legislators struggling with the adjustment of copyright law to a digital environment face many of these questions and concerns. Providers and users of digital information should also be aware of the issues. For more information about these topics, visit the Creative Commons Web site at http://creativecommons.org/.

FIGURES

- Table 1-4: Comparing an expedition to OOP

CLASSROOM ACTIVITIES

1. Quick Quizzes:
   1) Objects in an object-oriented system provide services by running _____.
      A. classes
      B. methods
      Answer: B
   2) A(n) _____ defines data resources and behavior for a set of objects.
      Answer: Class
   3) A(n) _____ is a rule of behavior.
      Answer: Method

2. Class Discussions:
   1) Separate the students into groups of five. Assign one student the job of porter, two the job of pathfinder, and two the job of cook. Have them go through the analogy the author points out in this section. You could also extend this into another analogy of a small business. Students could play the role of manager, data entry individuals, and accounting specialists.
   2) What is your responsibility as a user of digital media, such as music or software? How might you share this media? What are the consequences of violating media laws?
3. Lab Activity:
   1) Open a software program on your computer. See if you can find the copyright notices regarding its use and protection. Open two other programs and do the same. How do the copyright notices differ, and how are they the same? How might these copyright protections affect you if you were the programmer?

Key Terms

- **Application software:** Programs that allow human users to accomplish specialized tasks. (9)
- **Assembly language:** Uses mnemonic symbols to represent instructions and data. Also called second generation programming languages. (19)
- **Auxiliary input/output (I/O):** Devices such as printers and scanners. (6)
- **Auxiliary storage device:** Devices for long-term storage of data and programs, such as hard disks, DVDs, or flash memory. (6)
- **Bit:** A binary digit. (5)
- **Byte:** A sequence, consisting of 8 adjacent bits, used to encode character in memory. (5)
- **Central processing unit (CPU):** A major hardware component that consists of the arithmetic/logic unit and the control unit. (8)
- **Hardware:** The actual computing machine and its support devices. (5)
- **Information hiding:** The principle of providing access to services but not to data resources. (25)
- **Instance variables:** A list of data resources. (24)
- **Internal Memory:** Used for momentary storage of data and programs. Also called RAM. (6)
- **Machine language:** Uses only binary digits, 1 and 0, to code programs. Also called first generation programming languages. (19)
- **Network connection:** Used to connect to the Internet, other computers and the rest of the world. (6)
- **Object-oriented programming:** A programming process in which a program is subdivided into objects. (23)
- **Primary memory:** High-speed memory contained in the computer. Also called random access memory (RAM). (7)
- **RAM:** Stands for random access memory; high-speed internal memory; also called primary memory. (7)
- **Secondary memory:** Collective term for auxiliary storage devices where operating systems, applications, and documents are stored. Includes both hard disks and portable storage media such as DVDs and flash memory sticks. (7)
- **Software:** Programs that give computer hardware system useful functionality. (5)

- **Software development life cycle (SDLC):** The process that a program goes through. It consists of the development, maintenance, and demise of a software system. The phases include analysis, design, coding, testing/verification, maintenance, and obsolescence. (20)
- **System software:** The programs that allow users to write and execute other programs. For example, operating systems such as Unix, Linux, and MacOS. (9)
- **Ubiquitous computing:** The presence of computers, which are often hidden, in many items used in everyday life. (5)
- **User interface:** Supports moment-to-moment communication between a user and the computer. (6)
- **Waterfall model:** A version of the software development life cycle (SDLC) consisting of six phases in which the results of each phase flow down to the next. (20)

Projects to Assign

- In Project 1-1, students will discuss the different components of the computer that they are using for this course, including software, hardware, memory, CPU requirements (speed, etc.), and the operating system.
- In the Critical Thinking Activity, students will discuss the ethical and legal issues surrounding copyrighting.

Assess

- Administer the ExamView test for Chapter 1.

Top of Document