

---

## Chapter 3: Software

### Learning Objectives

Upon successful completion of this chapter, you will be able to:

- define the term *software*;
- describe the two primary categories of software;
- describe the role ERP software plays in an organization;
- describe cloud computing and its advantages and disadvantages for use in an organization; and
- define the term *open-source* and identify its primary characteristics.

### Introduction

The second component of an information system is software. Simply put: Software is the set of instructions that tell the hardware what to do. Software is created through the process of programming (we will cover the creation of software in more detail in chapter 10). Without software, the hardware would not be functional.

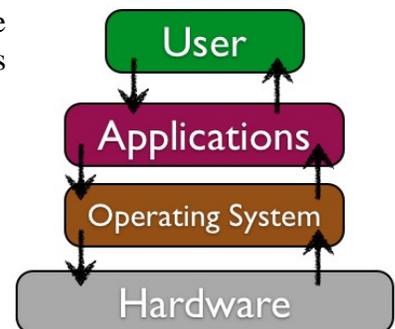
### Types of Software

Software can be broadly divided into two categories: operating systems and application software. *Operating systems* manage the hardware and create the interface between the hardware and the user. *Application software* is the category of programs that do something useful for the user.

### Operating Systems

The operating system provides several essential functions, including:

1. managing the hardware resources of the computer;
2. providing the user-interface components;
3. providing a platform for software developers to write applications.

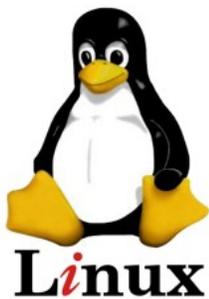


All computing devices run an operating system. For personal computers, the most popular operating systems are Microsoft's Windows, Apple's OS X, and different versions of Linux. Smartphones and tablets run operating systems as well, such as Apple's iOS, Google's Android, Microsoft's Windows Mobile, and Blackberry.

Early personal-computer operating systems were simple by today's standards; they did not provide multitasking and required the user to type commands to initiate an action. The amount of memory that early

operating systems could handle was limited as well, making large programs impractical to run. The most popular of the early operating systems was IBM's Disk Operating System, or DOS, which was actually developed for them by Microsoft.

In 1984, Apple introduced the Macintosh computer, featuring an operating system with a graphical user interface. Though not the first graphical operating system, it was the first one to find commercial success. In 1985, Microsoft released the first version of Windows. This version of Windows was not an operating system, but instead was an application that ran on top of the DOS operating system, providing a graphical environment. It was quite limited and had little commercial success. It was not until the 1990 release of Windows 3.0 that Microsoft found success with a graphical user interface. Because of the hold of IBM and IBM-compatible personal computers on business, it was not until Windows 3.0 was released that business users began using a graphical user interface, ushering us into the graphical-computing era. Since 1990, both Apple and Microsoft have released many new versions of their operating systems, with each release adding the ability to process more data at once and access more memory. Features such as multitasking, virtual memory, and voice input have become standard features of both operating systems.



A third personal-computer operating system family that is gaining in popularity is Linux (pronounced “linn-ex”). Linux is a version of the Unix operating system that runs on the personal computer. Unix is an operating system used primarily by scientists and engineers on larger minicomputers. These are very expensive computers, and software developer Linus Torvalds wanted to find a way to make Unix run on less expensive personal computers. Linux was the result. Linux has many variations and now powers a large percentage of web servers in the world. It is also an example of *open-source software*, a topic we will cover later in this chapter.

*Linux logo*  
(Copyright: Larry Ewing)

### Sidebar: Mac vs. Windows

Are you a Mac? Are you a PC? Ever since its introduction in 1984, users of the Apple Macintosh have been quite biased about their preference for the Macintosh operating system (now called OS X) over Microsoft's. When Microsoft introduced Windows, Apple sued Microsoft, claiming that they copied the “look and feel” of the Macintosh operating system. In the end, Microsoft successfully defended themselves.

Over the past few years, Microsoft and Apple have traded barbs with each other, each claiming to have a better operating system and software. While Microsoft has always had the larger market share (see sidebar), Apple has been the favorite of artists, musicians, and the technology elite. Apple also provides a lot of computers to elementary schools, thus gaining a following among the younger generation.

### Sidebar: Why Is Microsoft Software So Dominant in the Business World?

If you've worked in the world of business, you may have noticed that almost all of the computers run a version of Microsoft's Windows operating system. Why is this? On almost all college campuses, you see a preponderance of Apple Macintosh laptops. In elementary schools, Apple reigns as well. Why has this not extended into the business world?

As we learned in chapter 1, almost all businesses used IBM mainframe computers back in the 1960s and 1970s. These same businesses shied away from personal computers until IBM released the PC in 1981.

When executives had to make a decision about purchasing personal computers for their employees, they would choose the safe route and purchase IBM. The saying then was: “No one ever got fired for buying IBM.” So over the next decade, companies bought IBM personal computers (or those compatible with them), which ran an operating system called DOS. DOS was created by Microsoft, so when Microsoft released Windows as the next iteration of DOS, companies took the safe route and started purchasing Windows.

Microsoft soon found itself with the dominant personal-computer operating system for businesses. As the networked personal computer began to replace the mainframe computer as the primary way of computing inside businesses, it became essential for Microsoft to give businesses the ability to administer and secure their networks. Microsoft developed business-level server products to go along with their personal computer products, thereby providing a complete business solution. And so now, the saying goes: “No one ever got fired for buying Microsoft.”

## Application Software

The second major category of software is application software. Application software is, essentially, software that allows the user to accomplish some goal or purpose. For example, if you have to write a paper, you might use the application-software program Microsoft Word. If you want to listen to music, you might use iTunes. To surf the web, you might use Internet Explorer or Firefox. Even a computer game could be considered application software.

### The “Killer” App

ITEM	NO.	UNIT	COST
MUCK	4	12	5
RAKE	1	100	100
TOBACCO	25	48	1200
CUT	2	600	1200
TONER	4	300	1200
SNUFF	2	600	1200
SUBTOTAL			1315.00
9.75% TAX			128.50
TOTAL			14438.16

*VisiCalc running on an Apple II. (Public Domain)*

When a new type of digital device is invented, there are generally a small group of technology enthusiasts who will purchase it just for the joy of figuring out how it works. However, for most of us, until a device can actually do something useful we are not going to spend our hard-earned money on it. A “killer” application is one that becomes so essential that large numbers of people will buy a device just to run that application. For the personal computer, the killer application was the spreadsheet. In 1979, VisiCalc, the first personal-computer spreadsheet package, was introduced. It was an immediate hit and drove sales of the Apple II. It also solidified the value of the personal computer beyond the relatively small circle of technology geeks. When the IBM PC was released, another spreadsheet program, Lotus 1-2-3, was the killer app for business

users.

## Productivity Software

Along with the spreadsheet, several other software applications have become standard tools for the workplace. These applications, called productivity software, allow office employees to complete their daily work. Many times, these applications come packaged together, such as in Microsoft’s Office suite. Here is a list of these applications and their basic functions:

- **Word processing:** This class of software provides for the creation of written documents. Functions include the ability to type and edit text, format fonts and paragraphs, and add, move, and delete text throughout the document. Most modern word-processing programs also have the ability to add tables, images, and various layout and formatting features to the document. Word processors save their documents as electronic files in a variety of formats. By far, the most popular word-processing package is Microsoft Word, which saves its files in the DOCX format. This format can be read/written by many other word-processor packages.
- **Spreadsheet:** This class of software provides a way to do numeric calculations and analysis. The working area is divided into rows and columns, where users can enter numbers, text, or formulas. It is the formulas that make a spreadsheet powerful, allowing the user to develop complex calculations that can change based on the numbers entered. Most spreadsheets also include the ability to create charts based on the data entered. The most popular spreadsheet package is Microsoft Excel, which saves its files in the XLSX format. Just as with word processors, many other spreadsheet packages can read and write to this file format.
- **Presentation:** This class of software provides for the creation of slideshow presentations. Harkening back to the days of overhead projectors and transparencies, presentation software allows its users to create a set of slides that can be printed or projected on a screen. Users can add text, images, and other media elements to the slides. Microsoft's PowerPoint is the most popular software right now, saving its files in PPTX format.
- Some office suites include other types of software. For example, Microsoft Office includes Outlook, its e-mail package, and OneNote, an information-gathering collaboration tool. The professional version of Office also includes Microsoft Access, a database package. (Databases are covered more in chapter 4.)

Microsoft popularized the idea of the office-software productivity bundle with their release of Microsoft Office. This package continues to dominate the market and most businesses expect employees to know how to use this software. However, many competitors to Microsoft Office do exist and are compatible with the file formats used by Microsoft (see table below). Recently, Microsoft has begun to offer a web version of their Office suite. Similar to Google Drive, this suite allows users to edit and share documents online utilizing cloud-computing technology. Cloud computing will be discussed later in this chapter.

Category:	Word Processing	Spreadsheet	Presentation	Other
Suite:				
Microsoft Office	Word	Excel	PowerPoint	Outlook (email), Access (database), OneNote (information gathering)
Apple iWork	Pages	Numbers	Keynote	Integrates with iTunes, iCloud, and other Apple software
OpenOffice	Writer	Calc	Impress	Base (database), Draw (drawing), Math (equations)
Google Drive	Document	Spreadsheet	Presentation	Gmail (email), Forms (online form data collection), Draw (drawing)

*Comparison of office application software suites*

#### Utility Software and Programming Software

Two subcategories of application software worth mentioning are utility software and programming software. Utility software includes software that allows you to fix or modify your computer in some way. Examples include antivirus software and disk defragmentation software. These types of software packages were invented to fill shortcomings in operating systems. Many times, a subsequent release of an operating system will include these utility functions as part of the operating system itself.

Programming software is software whose purpose is to make more software. Most of these programs provide programmers with an environment in which they can write the code, test it, and convert it into the format that can then be run on a computer.

#### Sidebar: “PowerPointed” to Death

As presentation software, specifically Microsoft PowerPoint, has gained acceptance as the primary method to formally present information in a business setting, the art of giving an engaging presentation is becoming rare. Many presenters now just read the bullet points in the presentation and immediately bore those in attendance, who can already read it for themselves.

The real problem is not with PowerPoint as much as it is with the person creating and presenting. Author and thinker Seth Godin put it this way: “PowerPoint could be the most powerful tool on your computer. But it’s not. It’s actually a dismal failure. Almost every PowerPoint presentation sucks rotten eggs.”<sup>1</sup> The software used to help you communicate should not duplicate the presentation you want to give, but instead

1. From *Why are your PowerPoints so bad?* available for download at <http://www.sethgodin.com/freeprize/reallybad-1.pdf>.

it should support it. I highly recommend the book *Presentation Zen* by Garr Reynolds to anyone who wants to improve their presentation skills.

Software developers are becoming aware of this problem as well. New digital presentation technologies are being developed, with the hopes of becoming “the next PowerPoint.” One innovative new presentation application is Prezi. Prezi is a presentation tool that uses a single canvas for the presentation, allowing presenters to place text, images, and other media on the canvas, and then navigate between these objects as they present. Just as with PowerPoint, Prezi should be used to supplement the presentation. And we must always remember that sometimes the best presentations are made with no digital tools.

### Sidebar: I Own This Software, Right? Well . . .

When you purchase software and install it on your computer, are you the owner of that software? Technically, you are not! When you install software, you are actually just being given a license to use it. When you first install a software package, you are asked to agree to the terms of service or the license agreement. In that agreement, you will find that your rights to use the software are limited. For example, in the terms of the Microsoft Office Excel 2010 software license, you will find the following statement: “This software is licensed, not sold. This agreement only gives you some rights to use the features included in the software edition you licensed.”

For the most part, these restrictions are what you would expect: you cannot make illegal copies of the software and you may not use it to do anything illegal. However, there are other, more unexpected terms in these software agreements. For example, many software agreements ask you to agree to a limit on liability. Again, from Microsoft: “Limitation on and exclusion of damages. You can recover from Microsoft and its suppliers only direct damages up to the amount you paid for the software. You cannot recover any other damages, including consequential, lost profits, special, indirect or incidental damages.” What this means is that if a problem with the software causes harm to your business, you cannot hold Microsoft or the supplier responsible for damages.

#### Applications for the Enterprise

As the personal computer proliferated inside organizations, control over the information generated by the organization began splintering. Say the customer service department creates a customer database to keep track of calls and problem reports, and the sales department also creates a database to keep track of customer information. Which one should be used as the master list of customers? As another example, someone in sales might create a spreadsheet to calculate sales revenue, while someone in finance creates a different one that meets the needs of their department. However, it is likely that the two spreadsheets will come up with different totals for revenue. Which one is correct? And who is managing all of this information?

## Enterprise Resource Planning

In the 1990s, the need to bring the organization's information back under centralized control became more apparent. The enterprise resource planning (ERP) system (sometimes just called enterprise software) was developed to bring together an entire organization in one software application. Simply put, an ERP system is a software application utilizing a central database that is implemented throughout the entire organization. Let's take a closer look at this definition:

- “A software application”: An ERP is a software application that is used by many of an organization's employees.
- “utilizing a central database”: All users of the ERP edit and save their information from the data source. What this means practically is that there is only one customer database, there is only one calculation for revenue, etc.
- “that is implemented throughout the entire organization”: ERP systems include functionality that covers all of the essential components of a business. Further, an organization can purchase modules for its ERP system that match specific needs, such as manufacturing or planning.



*Registered trademark of SAP*

ERP systems were originally marketed to large corporations. However, as more and more large companies began installing them, ERP vendors began targeting mid-sized and even smaller businesses. Some of the more well-known ERP systems include those from SAP, Oracle, and Microsoft.

In order to effectively implement an ERP system in an organization, the organization must be ready to make a full commitment. All aspects of the organization are affected as old systems are replaced by the ERP system. In general, implementing an ERP system can take two to three years and several million dollars. In most cases, the cost of the software is not the most expensive part of the implementation: it is the cost of the consultants!

So why implement an ERP system? If done properly, an ERP system can bring an organization a good return on their investment. By consolidating information systems across the enterprise and using the software to enforce best practices, most organizations see an overall improvement after implementing an ERP. Business processes as a form of competitive advantage will be covered in chapter 9.

## Sidebar: Y2K and ERP

The initial wave of software-application development began in the 1960s, when applications were developed for mainframe computers. In those days, computing was expensive, so applications were designed to take as little space as possible. One shortcut that many programmers took was in the storage of dates, specifically the year. Instead of allocating four digits to hold the year, many programs allocated two digits, making the assumption that the first two digits were “19”. For example, to calculate how old someone was, the application would take the last two digits of the current year (for 1995, for example, that

would be “95”) and then subtract the two digits stored for the birthday year (“65” for 1965). 95 minus 65 gives an age of 30, which is correct.

However, as the year 2000 approached, many of these “legacy” applications were still being used, and businesses were very concerned that any software applications they were using that needed to calculate dates would fail. To update our age-calculation example, the application would take the last two digits of the current year (for 2012, that would be “12”) and then subtract the two digits stored for the birthday year (“65” for 1965). 12 minus 65 gives an age of -53, which would cause an error. In order to solve this problem, applications would have to be updated to use four digits for years instead of two. Solving this would be a massive undertaking, as every line of code and every database would have to be examined.

This is where companies gained additional incentive to implement an ERP system. For many organizations that were considering upgrading to ERP systems in the late 1990s, this problem, known as Y2K (year 2000), gave them the extra push they needed to get their ERP installed before the year 2000. ERP vendors guaranteed that their systems had been designed to be Y2K compliant – which simply meant that they stored dates using four digits instead of two. This led to a massive increase in ERP installations in the years leading up to 2000, making the ERP a standard software application for businesses.

#### Customer Relationship Management

A customer relationship management (CRM) system is a software application designed to manage an organization’s customers. In today’s environment, it is important to develop relationships with your customers, and the use of a well-designed CRM can allow a business to personalize its relationship with each of its customers. Some ERP software systems include CRM modules. An example of a well-known CRM package is Salesforce.

#### Supply Chain Management

Many organizations must deal with the complex task of managing their supply chains. At its simplest, a supply chain is the linkage between an organization’s suppliers, its manufacturing facilities, and the distributors of its products. Each link in the chain has a multiplying effect on the complexity of the process: if there are two suppliers, one manufacturing facility, and two distributors, for example, then there are  $2 \times 1 \times 2 = 4$  links to handle. However, if you add two more suppliers, another manufacturing facility, and two more distributors, then you have  $4 \times 2 \times 4 = 32$  links to manage.

A supply chain management (SCM) system manages the interconnection between these links, as well as the inventory of the products in their various stages of development. A full definition of a supply chain management system is provided by the Association for Operations Management: “The design, planning, execution, control, and monitoring of supply chain activities with the objective of creating net value, building a competitive infrastructure, leveraging worldwide logistics, synchronizing supply with demand, and measuring performance globally.”<sup>2</sup> [Most ERP systems include a supply chain management module.](#)

## Mobile Applications

Just as with the personal computer, mobile devices such as tablet computers and smartphones also have operating systems and application software. In fact, these mobile devices are in many ways just smaller versions of personal computers. A mobile app is a software application programmed to run specifically on a mobile device.

2. <http://www.apics.org/dictionary/dictionary-information?ID=3984>

As we saw in chapter 2, smartphones and tablets are becoming a dominant form of computing, with many more smartphones being sold than personal computers. This means that organizations will have to get smart about developing software on mobile devices in order to stay relevant.

These days, most mobile devices run on one of two operating systems: Android or iOS. Android is an open-source operating system purchased and supported by Google; iOS is Apple's mobile operating system. In the fourth quarter of 2012, Android was installed on 70.1% of all mobile phones shipped, followed by 21.0% for iOS. Other mobile operating systems of note are Blackberry (3.2%) and Windows (2.6%).<sup>3</sup>

As organizations consider making their digital presence compatible with mobile devices, they will have to decide whether to build a mobile app. A mobile app is an expensive proposition, and it will only run on one type of mobile device at a time. For example, if an organization creates an iPhone app, those with Android phones cannot run the application. Each app takes several thousand dollars to create, so this is not a trivial decision for many companies.

One option many companies have is to create a website that is mobile-friendly. A mobile website works on all mobile devices and costs about the same as creating an app. We will discuss the question of whether to build a mobile app more thoroughly in Chapter 10.

## Cloud Computing

Historically, for software to run on a computer, an individual copy of the software had to be installed on the computer, either from a disk or, more recently, after being downloaded from the Internet. The concept of "cloud" computing changes this, however.

To understand cloud computing, we first have to understand what the cloud is. "The cloud" refers to applications, services, and data storage on the Internet. These service providers rely on giant server farms and massive storage devices that are connected via Internet protocols. Cloud computing is the use of these services by individuals and organizations.

You probably already use cloud computing in some forms. For example, if you access your e-mail via your web browser, you are using a form of cloud computing. If you use Google Drive's applications, you are using cloud computing. While these are free versions of cloud computing, there is big business in providing applications and data storage over the web. Salesforce (see above) is a good example of cloud computing – their entire suite of CRM applications are offered via the cloud. Cloud computing is not limited to web applications: it can also be used for services such as phone or video streaming.

### Advantages of Cloud Computing

- No software to install or upgrades to maintain.
- Available from any computer that has access to the Internet.
- Can scale to a large number of users easily.
- New applications can be up and running very quickly.
- Services can be leased for a limited time on an as-needed basis.
- Your information is not lost if your hard disk crashes or your laptop is stolen.
- You are not limited by the available memory or disk space on your computer.

3. Taken from IDC Worldwide Mobile Phone Tracker, February 14, 2013. Full report available at <http://www.idc.com/getdoc.jsp?containerId=prUS23946013>

## Disadvantages of Cloud Computing

- Your information is stored on someone else’s computer – how safe is it?
- You must have Internet access to use it. If you do not have access, you’re out of luck.
- You are relying on a third-party to provide these services.

Cloud computing has the ability to really impact how organizations manage technology. For example, why is an IT department needed to purchase, configure, and manage personal computers and software when all that is really needed is an Internet connection?

## Using a Private Cloud

Many organizations are understandably nervous about giving up control of their data and some of their applications by using cloud computing. But they also see the value in reducing the need for installing software and adding disk storage to local computers. A solution to this problem lies in the concept of a *private cloud*. While there are various models of a private cloud, the basic idea is for the cloud service provider to section off web server space for a specific organization. The organization has full control over that server space while still gaining some of the benefits of cloud computing.

## Virtualization

One technology that is utilized extensively as part of cloud computing is “virtualization.” Virtualization is the process of using software to simulate a computer or some other device. For example, using virtualization, a single computer can perform the functions of several computers. Companies such as EMC provide virtualization software that allows cloud service providers to provision web servers to their clients quickly and efficiently. Organizations are also implementing virtualization in order to reduce the number of servers needed to provide the necessary services. For more detail on how virtualization works, [see this informational page from VMWare](#).

## Software Creation

How is software created? If software is the set of instructions that tells the hardware what to do, how are these instructions written? If a computer reads everything as ones and zeroes, do we have to learn how to write software that way?

Modern software applications are written using a programming language. A programming language consists of a set of commands and syntax that can be organized logically to execute specific functions. This language generally consists of a set of readable words combined with symbols. Using this language, a programmer writes a program (called the source code) that can then be compiled into machine-readable form, the ones and zeroes necessary to be executed by the CPU. Examples of well-known programming languages today include Java, PHP, and various flavors of C (Visual C, C++, C#). Languages such as HTML and Javascript are used to develop web pages. Most of the time, programming is done inside a programming environment; when you purchase a copy of Visual Studio from Microsoft, it provides you with an editor, compiler, and help for many of Microsoft’s programming languages.

Software programming was originally an individual process, with each programmer working on an entire program, or several programmers each working on a portion of a larger program. However, newer

methods of software development include a more collaborative approach, with teams of programmers working on code together. We will cover information-systems development more fully in chapter 10.

## Open-Source Software

When the personal computer was first released, it did not serve any practical need. Early computers were difficult to program and required great attention to detail. However, many personal-computer enthusiasts immediately banded together to build applications and solve problems. These computer enthusiasts were happy to share any programs they built and solutions to problems they found; this collaboration enabled them to more quickly innovate and fix problems.

As software began to become a business, however, this idea of sharing everything fell out of favor, at least with some. When a software program takes hundreds of man-hours to develop, it is understandable that the programmers do not want to just give it away. This led to a new business model of restrictive software licensing, which required payment for software, a model that is still dominant today. This model is sometimes referred to as *closed source*, as the source code is not made available to others.

There are many, however, who feel that software should not be restricted. Just as with those early hobbyists in the 1970s, they feel that innovation and progress can be made much more rapidly if we share what we learn. In the 1990s, with Internet access connecting more and more people together, the open-source movement gained steam.

Open-source software is software that makes the source code available for anyone to copy and use. For most of us, having access to the source code of a program does us little good, as we are not programmers and won't be able to do much with it. The good news is that open-source software is also available in a compiled format that we can simply download and install. The open-source movement has led to the development of some of the most-used software in the world, including the Firefox browser, the Linux operating system, and the Apache web server. Many also think open-source software is superior to closed-source software. Because the source code is freely available, many programmers have contributed to open-source software projects, adding features and fixing bugs.

Many businesses are wary of open-source software precisely because the code is available for anyone to see. They feel that this increases the risk of an attack. Others counter that this openness actually decreases the risk because the code is exposed to thousands of programmers who can incorporate code changes to quickly patch vulnerabilities.

There are many arguments on both sides of the aisle for the benefits of the two models. Some benefits of the open-source model are:

- The software is available for free.
- The software source-code is available; it can be examined and reviewed before it is installed.
- The large community of programmers who work on open-source projects leads to quick bug-fixing and feature additions.

Some benefits of the closed-source model are:

- By providing financial incentive for software development, some of the brightest minds have chosen software development as a career.
- Technical support from the company that developed the software.

Today there are thousands of open-source software applications available for download. For example, as we discussed previously in this chapter, you can get the productivity suite from Open Office. One good place to search for open-source software is [sourceforge.net](http://sourceforge.net), where thousands of software applications are available for free download.

## Summary

Software gives the instructions that tell the hardware what to do. There are two basic categories of software: operating systems and applications. Operating systems provide access to the computer hardware and make system resources available. Application software is designed to meet a specific goal. Productivity software is a subset of application software that provides basic business functionality to a personal computer: word processing, spreadsheets, and presentations. An ERP system is a software application with a centralized database that is implemented across the entire organization. Cloud computing is a method of software delivery that runs on any computer that has a web browser and access to the Internet. Software is developed through a process called programming, in which a programmer uses a programming language to put together the logic needed to create the program. While most software is developed using a closed-source model, the open-source movement is gaining more support today.

## Study Questions

1. Come up with your own definition of software. Explain the key terms in your definition.
2. What are the functions of the operating system?
3. Which of the following are operating systems and which are applications: Microsoft Excel, Google Chrome, iTunes, Windows, Android, Angry Birds.
4. What is your favorite software application? What tasks does it help you accomplish?
5. What is a “killer” app? What was the killer app for the PC?
6. How would you categorize the software that runs on mobile devices? Break down these apps into at least three basic categories and give an example of each.
7. Explain what an ERP system does.
8. What is open-source software? How does it differ from closed-source software? Give an example of each.
9. What does a software license grant?
10. How did the Y2K (year 2000) problem affect the sales of ERP systems?

## Exercises

1. Go online and find a case study about the implementation of an ERP system. Was it successful? How long did it take? Does the case study tell you how much money the organization spent?

2. What ERP system does your university or place of employment use? Find out which one they use and see how it compares to other ERP systems.
3. If you were running a small business with limited funds for information technology, would you consider using cloud computing? Find some web-based resources that support your decision.
4. Download and install [Open Office](#). Use it to create a document or spreadsheet. How does it compare to Microsoft Office? Does the fact that you got it for free make it feel less valuable?
5. Go to [sourceforge.net](http://sourceforge.net) and review their most downloaded software applications. Report back on the variety of applications you find. Then pick one that interests you and report back on what it does, the kind of technical support offered, and the user reviews.
6. Review [this article](#) on the security risks of open-source software. Write a short analysis giving your opinion on the different risks discussed.
7. What are three examples of programming languages? What makes each of these languages useful to programmers?