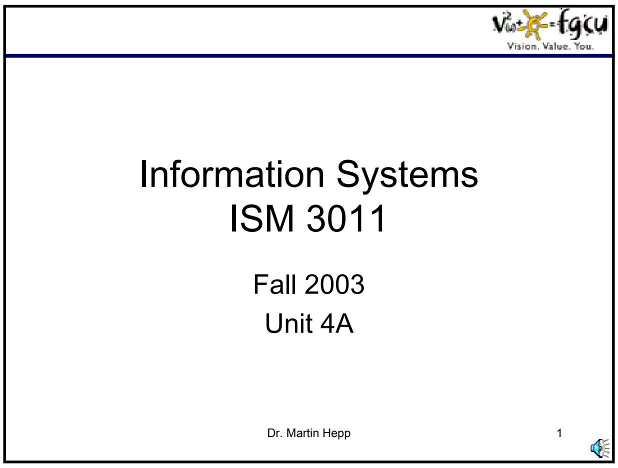
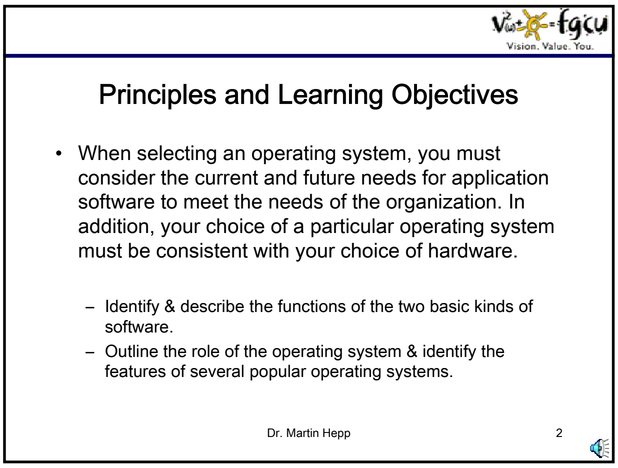


Information Systems ISM 3011

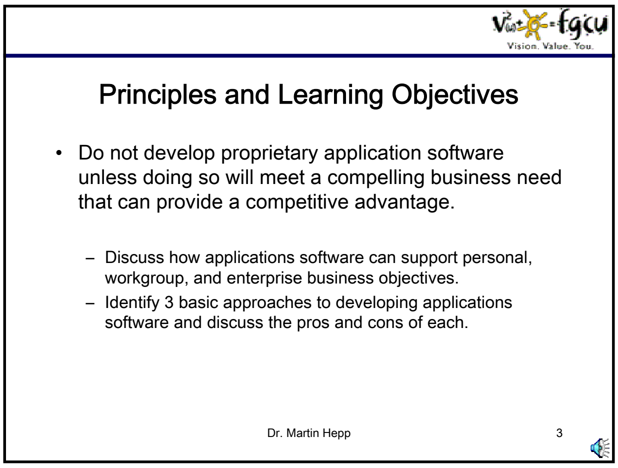
Fall 2003

Unit 4A



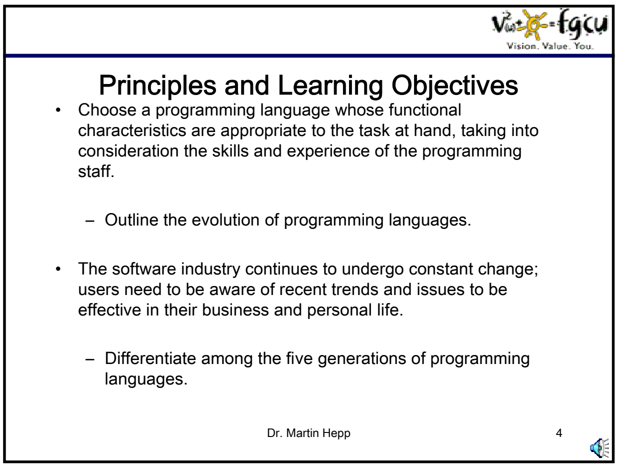
Principles and Learning Objectives

- When selecting an operating system, you must consider the current and future needs for application software to meet the needs of the organization. In addition, your choice of a particular operating system must be consistent with your choice of hardware.
  - Identify & describe the functions of the two basic kinds of software.
  - Outline the role of the operating system & identify the features of several popular operating systems.



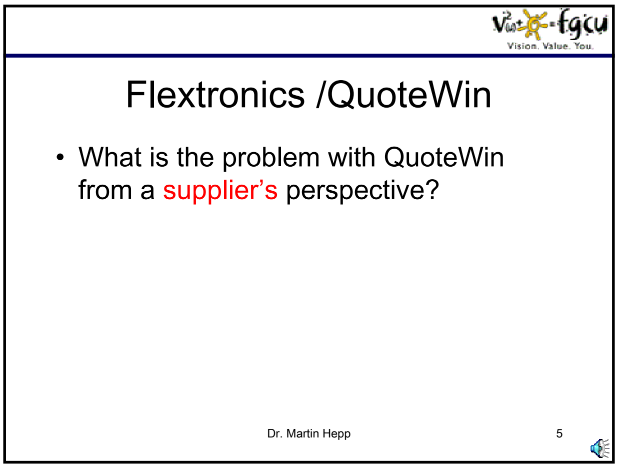
Principles and Learning Objectives

- Do not develop proprietary application software unless doing so will meet a compelling business need that can provide a competitive advantage.
  - Discuss how applications software can support personal, workgroup, and enterprise business objectives.
  - Identify 3 basic approaches to developing applications software and discuss the pros and cons of each.



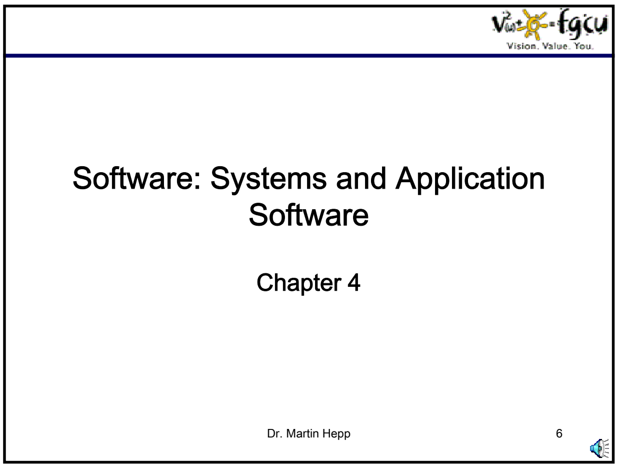
Principles and Learning Objectives

- Choose a programming language whose functional characteristics are appropriate to the task at hand, taking into consideration the skills and experience of the programming staff.
  - Outline the evolution of programming languages.
- The software industry continues to undergo constant change; users need to be aware of recent trends and issues to be effective in their business and personal life.
  - Differentiate among the five generations of programming languages.



Flextronics /QuoteWin

- What is the problem with QuoteWin from a supplier’s perspective?



Software: Systems and Application Software

Chapter 4

An Overview of Software

- **Computer program** - sequences of instructions for the computer
- **Documentation** - describes program functions
- **Systems software** - coordinates the activities of hardware & programs
- **Applications software** - helps users solve particular problems

The Importance of Software in Business

FIGURE 4.1

The Importance of Software in Business

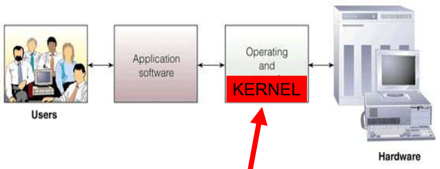
Since the 1950s, businesses have greatly increased their expenditures on software as compared with hardware.

Operating Systems

Operating Systems

- Perform common hardware functions
- Provide a user interface
- Provide hardware independence
- Manage system memory
- Manage processing
- Control access to system resources
- Manage files

The Role of the Operating System



KERNEL

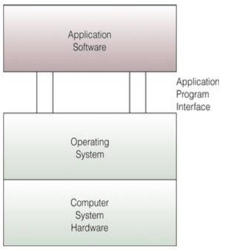
FIGURE 4.2

The role of the operating system and other systems software is as an interface or buffer between application software and hardware.

Hardware Independence

FIGURE 4.3

Application Program Interface Links Application Software to the Operating System



Examples:

•Accessing Printers (Windows-API) or

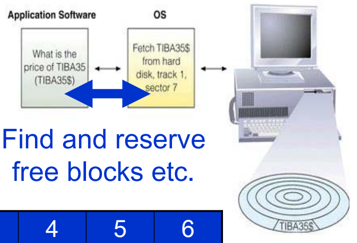
•Scanners and Digital Cameras (TWAIN)

Memory Management

FIGURE 4.4

An Example of the Operating System Controlling Physical Access to Data

The user prompts the application software for specific data. The operating system translates this prompt into instructions for the hardware, which finds the data the user requested. Having successfully completed this task, the operating system then relays the data back to the user via the application software.


Application Software
OS
What is the price of TIBA35 (TIBA35$)
Fetch TIBA35$ from hard disk, track 1, sector 7
TIBA35$

Find and reserve free blocks etc.

1	2	3	4	5	6

Memory segments

Processing Tasks

- **Multitasking -** more than one program (task) can run at a time using a single processor
- **Multi-User OS -** multiple users can simultaneously use the resources of a single processor
- **Scalability -** easy adaptation to more users or tasks

System Speed and RAM

- Why does a bigger primary storage capacity (more RAM) increase your PCs speed?
- Does more memory always lead to a higher system performance?

Off-the-Shelf Software

- The book says on page 147 that one should check whether the software manufacturer is financially solvent and reliable. Why?

Enterprise Operating Systems

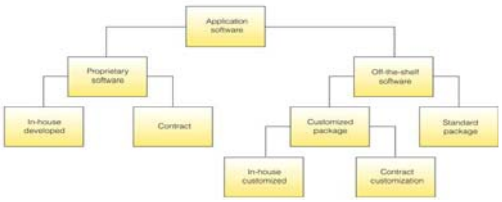
- IBM's OS/390
- HP's MPE/iX (Multiprogramming Executive with integrated POSIX)
- IBM's z/OS
- Linux

Sources of Software

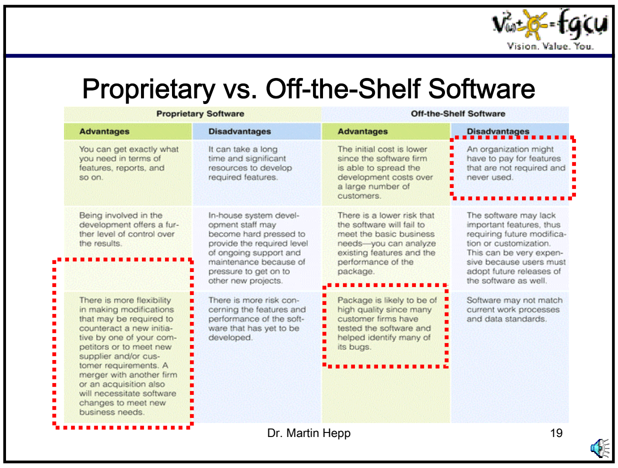
FIGURE 4.6

Sources of Software: Proprietary and Off-the-Shelf

Some off-the-shelf software may be modified to allow some customization.



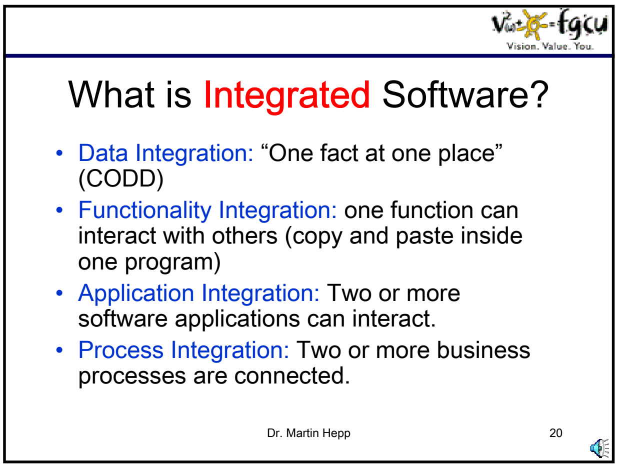
Some content with kind permission of Thomson Course Technology



Proprietary vs. Off-the-Shelf Software

Proprietary Software		Off-the-Shelf Software	
Advantages	**Disadvantages**	**Advantages**	**Disadvantages**
You can get exactly what you need in terms of features, reports, and so on.	It can take a long time and significant resources to develop required features.	The initial cost is lower since the software firm is able to spread the development costs over a large number of customers.	An organization might have to pay for features that are not required and never used.
Being involved in the development offers a further level of control over the results.	In-house system development staff may become hard pressed to provide the required level of ongoing support and maintenance because of pressure to get on to other new projects.	There is a lower risk that the software will fail to meet the basic business needs—you can analyze existing features and the performance of the package.	The software may lack important features, thus requiring future modification or customization. This can be very expensive because users must adopt future releases of the software as well.
There is more flexibility in making modifications that may be required to counteract a new initiative by one of your competitors or to meet new supplier and/or customer requirements. A merger with another firm or an acquisition also will necessitate software changes to meet new business needs.	There is more risk concerning the features and performance of the software that has yet to be developed.	Package is likely to be of high quality since many customer firms have tested the software and helped identify many of its bugs.	Software may not match current work processes and data standards.

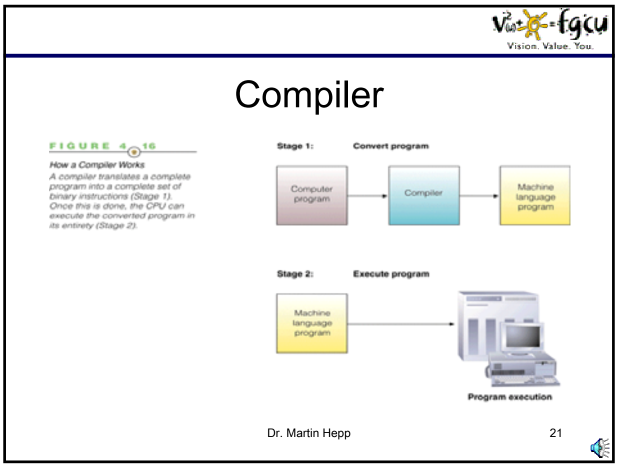
Dr. Martin Hepp 19



What is Integrated Software?

- Data Integration: “One fact at one place” (CODD)
- Functionality Integration: one function can interact with others (copy and paste inside one program)
- Application Integration: Two or more software applications can interact.
- Process Integration: Two or more business processes are connected.

Dr. Martin Hepp 20



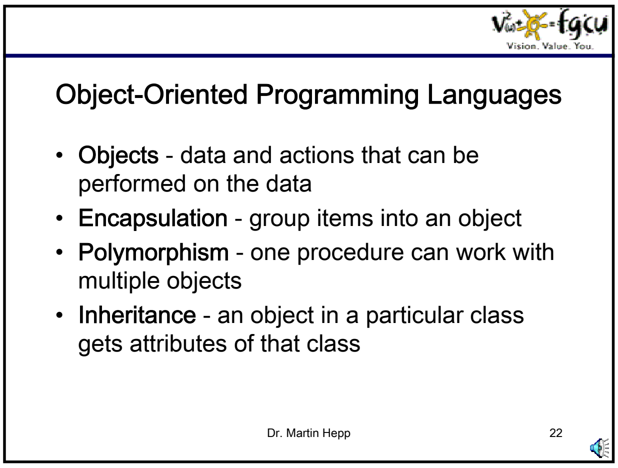
Compiler

FIGURE 4.16

How a Compiler Works

A compiler translates a complete program into a complete set of binary instructions (Stage 1). Once this is done, the CPU can execute the converted program in its entirety (Stage 2).

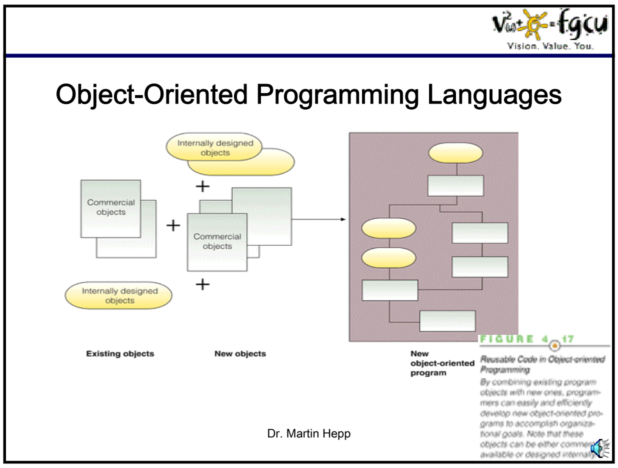
Dr. Martin Hepp 21



Object-Oriented Programming Languages

- **Objects** - data and actions that can be performed on the data
- **Encapsulation** - group items into an object
- **Polymorphism** - one procedure can work with multiple objects
- **Inheritance** - an object in a particular class gets attributes of that class

Dr. Martin Hepp 22



Object-Oriented Programming Languages

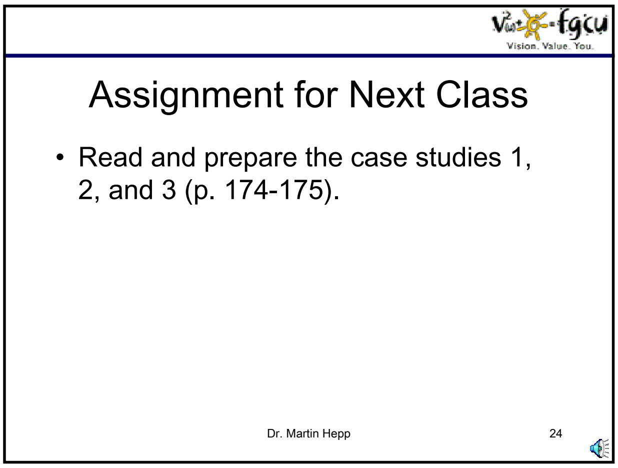
Internally designed objects
Commercial objects
+
Commercial objects
+
Internally designed objects
+
Existing objects
New objects
New object-oriented program

FIGURE 4.17

Reusable Code in Object-oriented Programming

By combining existing program objects with new ones, programmers can easily and efficiently develop new object-oriented programs to accomplish organizational goals. Note that these objects can be either comme available or designed internally.

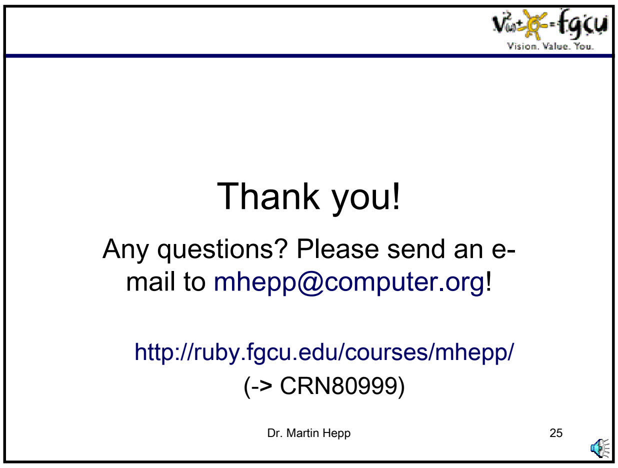
Dr. Martin Hepp



Assignment for Next Class

- Read and prepare the case studies 1, 2, and 3 (p. 174-175).

Dr. Martin Hepp 24



Thank you!

Any questions? Please send an e-mail to mhepp@computer.org!

http://ruby.fgcu.edu/courses/mhepp/
(-> CRN80999)

Dr. Martin Hepp 25