Chapter 12

Information system

Software that helps the user organize and analyze data

Electronic spreadsheets and database management systems

Software tools that allow the user to organize, manage, and analyze data in various ways **Spreadsheet**

A software application that allows the user to organize and analyze data using a grid of labeled **cells**

- A cell can contain data or a formula that is used to calculate a value
- Data stored in a cell can be text, numbers, or “special” data such as dates
- Spreadsheet cells are referenced by their row and column designation

The power of spreadsheets comes from the formulas that we can create and store in cells

- When a formula is stored in a cell, the *result* of the formula is displayed in the cell
- If we’ve set up the spreadsheet correctly, we could
- add or remove tutors.
- add additional weeks of data.
- change any of the data we have already stored and the corresponding calculations would automatically be updated.
- Formulas make use of basic arithmetic operations using the standard symbols (+, -, 2, *, and /)
- **Spreadsheet functions**

- Computations provided by the spreadsheet software that can be incorporated into formulas
- Range
- A set of contiguous cells specified by the endpoints

Function	Computes
SUM(val1, val2, ...) SUM(range)	Sum of the specified set of values
COUNT(val1, val2, ...) COUNT(range)	Count of the number of cells that contain values
MAX(val1, val2, ...) MAX(range)	Largest value from the specified set of values
SIN(angle)	The sine of the specified angle
PI()	The value of PI
STDEV(val1, val2, ...) STDEV(range)	The standard deviation from the specified sample values
TODAY()	Today's date
LEFT(text, num_chars)	The leftmost characters from the specified text
IF(test, true_val, false_val)	If the test is true, it returns the true_val; otherwise, it returns the false_val
ISBLANK (value)	Returns true if the specified value refers to an empty cell

Circular reference

A set of formulas that ultimately rely on each other

Cell	Contents
A1	=B7*COUNT(F8..K8)
B7	=A14+SUM(E40..E50)
E45	=G18+G19–D13
D13	=D12/A1

Possible tasks a spreadsheet could perform:

- Track sales
- Analyze sport statistics
- Maintain student grades
- Keep a car maintenance log

- Record and summarize travel expenses
- Track project activities and schedules
- Plan stock purchases
- Spreadsheets are also useful because of their *dynamic nature*, which provides the powerful ability to do *what-if analysis*

- **Database**

- A structured set of data

- **Database management system** (DBMS)

- A combination of software and data, made up of a physical database, a database engine, and a database schema

- **Physical database**

- A collection of files that contain the data

- **Database engine**

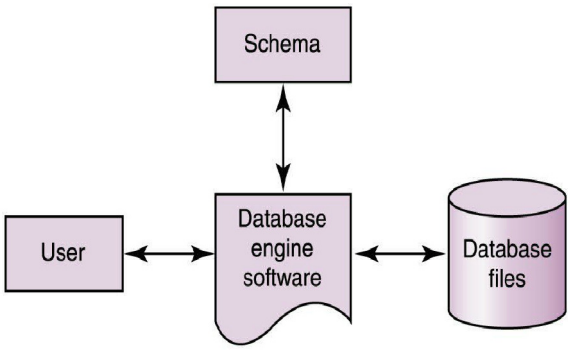
- Software that supports access to and modification of the database contents

- **Database schema**

- A specification of the logical structure of the data stored in the database

- **Database query**

- A request to retrieve data from a database



Relational DBMS

A DBMS in which the data items and the relationships among them are organized into **tables Tables**

A collection of **records**

Records (object, entity)

A collection of related **fields** that make up a single database entry

Fields (attributes)

A single value in a database record

Key

One or more fields of a database record that uniquely identifies it among all other records in the table We can express the schema for this part of the database as follows:

Movie (MovieId:key, Title, Genre, Rating)

Structured Query Language (SQL)

A comprehensive relational database language for data manipulation and queries

select *attribute-list* from *table-list*

where *condition* name of field

name of table value restriction

select Title from Movie where Rating = 'PG'

Result is a table containing all PG movies in table Movie

Entity-relationship (ER) modeling

A popular technique for designing relational databases

ER Diagram

A graphical representation of an ER model

Cardinality constraint

The number of relationships that may exist at one time among entities in an ER diagram

Electronic commerce

The process of buying and selling products and services using the Web.

Chapter 16

The Web

An infrastructure of information combined and the network software used to access it **Web page**

A document that contains or references various kinds of data

Links A connection between one web page and another

Website

A collection of related web pages

Web browser

A software tool that retrieves and displays eb pages

Web server

A computer set up to respond to requests for web pages

Uniform Resource Locator (URL)

A standard way of specifying the location of a Web page, containing the hostname, "/", and a file

Instant messaging (IM)

An application that allows people to send and receive messages in real time

- Both sender and receiver must have an IM running
- Most IM applications use a proprietary protocol that dictates the precise format and structure of the messages sent
- Instant messages are **not** secure

Blog or Weblog

- An online journal or newsletter that is frequently updated and intended for public consumption

Cookie

A small text file that a web server stores on your local computer's hard disk

- A cookie contains information about your visit to the site
- Cookies can be used

- to determine number of unique visitors to the site
- to customize the site for future visits
- to implement shopping carts that can be maintained from visit to visit
- Cookies are **not** dangerous
- **Hypertext Markup Language (HTML)**
- The language used to create or build a Web page
- **Markup language**
- A language that uses tags to annotate the information in a document

Tags

- The syntactic element in a markup language that indicates how information should be displayed
- Tags are enclosed in angle brackets (<. . . >)
- Words such as HEAD, TITLE, and BODY are called **elements** and specify the type of the tag
- Tags are often used in pairs, with a start tag such as <BODY> and a corresponding end tag with a / before the element name, such as </BODY>

The browser determines how the page should be displayed based on the tags

The browser

- Ignores the way we format the HTML document using carriage returns, extra spaces, and blank lines
- Takes into account the width and height of the browser window
- Reformats the contents to fit your browser window

Attribute

- Part of a tag that provides additional information about the element
- attribute-name = value
- <IMG SRC = "myPicture.gif">
- inserts the image stored in file "myPicture.gif"

Java applet

- A Java program designed to be embedded into an HTML document, transferred over the Web, and executed in a browser

JSP Scriptlet

A portion of code embedded in an HTML document designed to dynamically contribute to the content of the web page

Java Server Page

A web page that has a JSP scriptlet interwoven among the HTML content

A JSP scriptlet is encased in special tags beginning with <% and ending with %>
Imagine JSP scriptlets as having the expressive power of a full programming language

JSPs are executed on the server side where the web page resides
By the time it arrives at your computer, all active processing has taken place, producing a static (though dynamically created) web page
JSPs are particularly good for coordinating the interaction between a web page and an underlying database

Extensible Markup Language (XML)
A language that allows the user to describe the content of a document

- HTML describes how a document should look
- XML describes a document's meaning **Metalanguage**

A language for talking about, or defining, other languages
XML is a metalanguage

XML represents a standard format for organizing data without tying it to any particular type of output

Extensible Stylesheet Language (or XSL)

A language for defining transformations from XML documents to other output formats

Chapter 2

Natural Numbers

Zero and any number obtained by repeatedly adding one to it.

Examples: 100, 0, 45645, 32

Negative Numbers

A value less than 0, with a – sign

Examples: -24, -1, -45645, -32

Integers

A natural number, a negative number, zero

Examples: 249, 0, - 45645, - 32

Rational Numbers

An integer or the quotient of two integers

Examples: -249, -1, 0, 3/7, -2/5

How many ones are there in 642?

642 is 600 + 40 + 2 in **BASE 10**

The **base** of a number determines the number of digits and the value of digit positions

Continuing with our example…
642 in base 10 *positional notation* is:

$$\begin{aligned} 6 \times 10^2 &= 6 \times 100 = 600 \\ +\ 4 \times 10^1 &= 4 \times 10 = 40 \\ +\ 2 \times 10^0 &= 2 \times 1 = 2 = 642 \text{ in base } 10 \end{aligned}$$

What if 642 has the base of 13?

$$\begin{aligned} +\ 6 \times 13^2 &= 6 \times 169 = 1014 \\ +\ 4 \times 13^1 &= 4 \times 13 = 52 \\ +\ 2 \times 13^0 &= 2 \times 1 = 2 \\ &= 1068 \text{ in base } 10 \end{aligned}$$

642 in base 13 is equivalent to 1068 in base 10

Decimal is base 10 and has 10 digits: 0,1,2,3,4,5,6,7,8,9
Binary is base 2 and has 2 digits: 0,1

With distinct symbols for 10 and above.
Base 16 has 16 digits:
0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F

What is the decimal equivalent of the hexadecimal number DEF?

$$
\begin{aligned}
D \times 16^2 &= 13 \times 256 = 3328 \\
+ E \times 16^1 &= 14 \times 16 = 224 \\
+ F \times 16^0 &= 15 \times 1 = 15 \\
&= 3567 \text{ in base } 10
\end{aligned}
$$

What is the decimal equivalent of the octal number 642?

$$
\begin{aligned}
6 \times 8^2 &= 6 \times 64 = 384 \\
+ 4 \times 8^1 &= 4 \times 8 = 32 \\
+ 2 \times 8^0 &= 2 \times 1 = 2 \\
&= 418 \text{ in base } 10
\end{aligned}
$$

What is the decimal equivalent of the hexadecimal number DEF?

$$
\begin{aligned}
D \times 16^2 &= 13 \times 256 = 3328 \\
+ E \times 16^1 &= 14 \times 16 = 224 \\
+ F \times 16^0 &= 15 \times 1 = 15 \\
&= 3567 \text{ in base } 10
\end{aligned}
$$

What is the decimal equivalent of the binary number 1101110?

$$
\begin{aligned}
1 \times 2^6 &= 1 \times 64 = 64 \\
+ 1 \times 2^5 &= 1 \times 32 = 32 \\
+ 0 \times 2^4 &= 0 \times 16 = 0 \\
+ 1 \times 2^3 &= 1 \times 8 = 8 \\
+ 1 \times 2^2 &= 1 \times 4 = 4 \\
+ 1 \times 2^1 &= 1 \times 2 = 2 \\
+ 0 \times 2^0 &= 0 \times 1 = 0 \\
&= 110 \text{ in base } 10
\end{aligned}
$$

Converting Binary to Octal

- Mark groups of *three* (from right)
- Convert each group

10101011 → 10 101 011 → 2 5 3

10101011 is 253 in base 8

Converting Binary to Hexadecimal

- Mark groups of *four* (from right)
- Convert each group

10101011 → 1010 1011 → A B

10101011 is AB in base 16

Converting Decimal to Octal

What is 1988 (base 10) in base 8?

```
   248      31       3       0
8 1988   8 248    8 31    8 3
  16       24       24      0
  38       08        7      3
  32        8
  68        0
  64
   4
```

Answer is : **3 7 0 4**

Converting Decimal to Hexadecimal *What is 3567 (base 10) in base 16?*

```
    222      13       0
16 3567  16 222   16 13
   32       16        0
   36       62       13
   32       48
   47       14
   32
   15
```

Byte
8 bits

The number of bits in a word determines the word length of the computer, but it is usually a multiple of 8

- 32-bit machines
- 64-bit machines etc.

Chapter 3

Computers are **multimedia** devices,dealing with a vast array of information categories
Computers store, present, and help us modify

- Numbers
- Text
- Audio
- Images and graphics
- Video

Data compression
Reduction in the amount of space needed to store a piece of data
Compression ratio
The size of the compressed data divided by the size of the original data
A data compression technique can be
lossless, which means the data can be retrieved without any loss of the original information **lossy**, which means some information may be lost in the process of compaction

Information can be represented in one of two ways: **analog** or **digital**
Analog data
A continuous representation, analogous to the actual information it represents **Digital data**
A discrete representation, breaking the information up into separate elements

Computers cannot work well with analog data, so we digitize the data **Digitize**
Breaking data into pieces and representing those pieces separately
Why do we use binary to represent digitized data?

Important facts about electronic signals

- An analog signal continually fluctuates in voltage up and down
- A digital signal has only a high or low state, corresponding to the two binary digits
- All electronic signals (both analog and digital) degrade as they move down a line
- The voltage of the signal fluctuates due to environmental effects

Signed-magnitude number representation
The sign represents the ordering, and the digits represent the magnitude of the number

ASCII stands for American Standard Code for Information Interchange

ASCII originally used seven bits to represent each character, allowing for 128 unique characters Later extended ASCII evolved so that all eight bits were used

The first 32 characters in the ASCII character chart do not have a simple character representation to print to the screen

The Unicode Character Set

Extended ASCII is not enough for international use
One Unicode mapping uses 16 bits per character
How many characters can this mapping represent?
Unicode is a superset of ASCII
The first 256 characters correspond exactly to the extended ASCII character set

Assigning 16 bits to each character in a document uses too much file space
We need ways to store and transmit text efficiently
Text
compression
techniques
keyword
encoding run-
length encoding
Huffman encoding

Audio Formats

Audio Formats

- WAV, AU, AIFF, VQF, and MP3 MP3 (MPEG-2, audio layer 3 file) is dominant
- analyzes the frequency spread and discards information that can't be heard by humans
- bit stream is compressed using a form of Huffman encoding to achieve additional compression
- Color
- Perception of the frequencies of light that reach the retinas of our eyes
- Retinas have three types of color photoreceptor cone cells that correspond to the colors of red, green, and blue

color depth
The amount of data that is used to represent a color
HiColor
A 16-bit color depth: five bits used for each number in an RGB value with the extra bit sometimes used to represent transparency
TrueColor
A 24-bit color depth: eight bits used for each number in an RGB value

Digitizing a picture
Representing it as a collection of individual dots called pixels
Resolution

The number of pixels used to represent a picture
Raster Graphics
Storage of data on a pixel-by-pixel basis
Bitmap (BMP), GIF, JPEG, and PNG are raster-grahics formats

Bitmap format
Contains the pixel color values of the image from left to right and from top to bottom
GIF format (indexed color)
Each image is made up of only 256 colors
JPEG format
Averages color hues over short distances
PNG format
Like GIF but achieves greater compression with wider range of color depths *Which is better for line drawings? Pictures?*

Vector Graphics
Vector graphics
A format that describes an image in terms of lines and geometric shapes
A vector graphic is a series of commands that describe a line's direction, thickness, and color. The file sizes tend to be smaller because not every pixel is described.

Vector graphics can be resized mathematically and changes can be calculated dynamically as needed. Vector graphics are *not* good for representing real-world images

Representing Video

Video codec COmpressor/DECompressor Methods used to shrink the size of a movie to allow it to be played on a computer or over a network

Almost all video codecs use lossy compressions to minimize the huge amounts of data associated with video

Temporal compression

A technique based on differences between consecutive frames: If most of an image in two frames hasn't changed, why should we waste space to duplicate all of the similar information?

Spatial compression

A technique based on removing redundant information within a frame: This problem is essentially the same as that faced when compressing still images

Chapter 4

Computers and Electricity

Gate

A device that performs a basic operation on electrical signals

Circuits

Gates combined to perform more complicated tasks

Boolean expressions

Uses Boolean algebra, a mathematical notation for expressing two-valued logic Logic diagrams

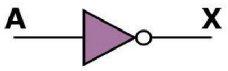
A graphical representation of a circuit; each gate has its own symbol Truth tables

A table showing all possible input values and the associated output values

Boolean Expression

$X = A'$

Logic Diagram Symbol



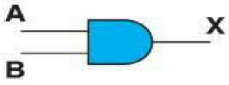
Truth Table

A	X
0	1
1	0

Boolean Expression

$X = A \cdot B$

Logic Diagram Symbol



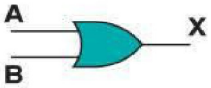
Truth Table

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Boolean Expression

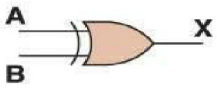
$X = A + B$

Logic Diagram Symbol

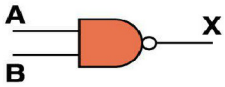


Truth Table

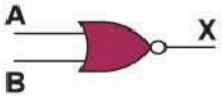
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Boolean Expression	Logic Diagram Symbol	Truth Table
$X = A \oplus B$		

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Boolean Expression	Logic Diagram Symbol	Truth Table
$X = (A \cdot B)'$		

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Boolean Expression	Logic Diagram Symbol	Truth Table
$X = (A + B)'$		

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Constructing Gates

Transistor

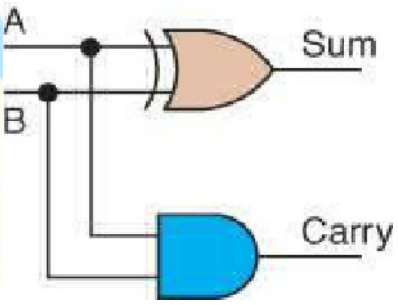
A device that acts either as a wire that conducts electricity or as a resistor that blocks the flow of electricity, depending on the voltage level of an input signal

A transistor has no moving parts, yet acts like a switch

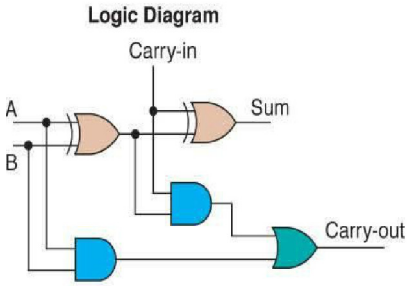
It is made of a semiconductor material, which is neither a particularly good conductor of electricity nor a particularly good insulator

Half adder

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Full adder



Truth Table

A	B	Carry-in	Sum	Carry-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Multiplexer

A circuit that uses a few input control signals to determine which of several output data lines is routed to its output

Integrated circuits (IC) are classified by the number of gates contained in them

Abbreviation	Name	Number of Gates
SSI	Small-Scale Integration	1 to 10
MSI	Medium-Scale Integration	10 to 100
LSI	Large-Scale Integration	100 to 100,000
VLSI	Very-Large-Scale Integration	more than 100,000

Chapter 7

Problem solving
The act of finding a solution to a perplexing, distressing, vexing, or unsettled question

How do you solve problems?
Understand the problem
Devise a plan
Carry out the plan
Look back

Ask questions!
Ask questions! Never reinvent the wheel!
Divide and Conquer!

Algorithms
Algorithm
A set of unambiguous instructions for solving a problem or subproblem in a finite amount of time using a finite amount of *data*
Abstract Step
An algorithmic step containing unspecified details
Concrete Step

An algorithm step in which all details are specified

Two methodologies used to develop computer solutions to a problem

- Top-down design focuses on the **tasks** to be done
- Object-oriented design focuses on the **data** involved in the solution (We will discuss this design in Ch. 9)

Summary of Methodology

Analyze the
Problem

Understand the
problem!!

Develop a plan of attack

List the Main Tasks (becomes Main Module)

Restate problem as a list of tasks (modules)

Give each task a name

Write the Remaining Modules

Restate each abstract module as a list of tasks

Give each task a name

Re-sequence and Revise as Necessary

Process ends when all steps (modules) are concrete

Control structure

An instruction that determines the order in which other instructions in a program are executed *Can you name the ones we defined in the functionality of pseudocode?*

Arrays

As data is being read into an array, a counter is updated so that we always know how many data items were stored

Sorted Arrays

The values stored in an array have unique keys of a type for which the relational operators are defined

Sorting rearranges the elements into either ascending or descending order within the array A sorted array is one in which the elements are in order

Sorting

Arranging items in a collection so that there is an ordering on one (or more) of the fields in the items **Sort Key**

The field (or fields) on which the ordering is based

Sorting algorithms

Algorithms that order the items in the collection based on the sort key

Selection Sort

A slight adjustment to this manual approach does away with the need to duplicate space

- As you cross a name off the original list, a free space opens up

- Instead of writing the value found on a second list, exchange it with the value currently in the position where the crossed-off item should go

Bubble Sort

Bubble Sort uses the same strategy:

Find the next item

Put it into its proper place

But uses a different scheme for finding the next item

Starting with the last list element, compare successive pairs of elements, swapping whenever the bottom element of the pair is smaller than the one above it

Bubble sort is very slow!

Insertion Sort

If you have only one item in the array, it is already sorted.

If you have two items, you can compare and swap them if necessary, sorting the first two with respect to themselves.

Take the third item and put it into its place relative to the first two

Now the first three items are sorted with respect to one another

Subprogram Statements

We can give a section of code a name and use that name as a statement in another part of the program When the name is encountered, the processing in the other part of the program halts while the named code is executed

What if the subprogram needs data from the calling unit?
Parameters
Identifiers listed in parentheses beside the subprogram declaration; sometimes called **formal parameters**

Arguments
Identifiers listed in parentheses on the subprogram call; sometimes called **actual parameters**

Recursion
The ability of a subprogram to call itself
Base case
The case to which we have an answer
General case
The case that expresses the solution in terms of a call to itself with a smaller version of the problem

Important Threads
Information Hiding
The practice of hiding the details of a module with the goal of controlling access to it **Abstraction**
A model of a complex system that includes only the details essential to the viewer
Information Hiding and **Abstraction** are two sides of the same coin
Data abstraction

Separation of the logical view of data from their implementation

Procedural abstraction

Separation of the logical view of actions from their implementation

Control abstraction

Separation of the logical view of a control structure from its implementation **Identifiers**

Names given to data and actions, by which we access the data and execute the actions

Read firstName, Set count to count + 1

Split(splitVal)

Giving names to data and actions is a form of abstraction

Chapter 9

Object-oriented Design

A problem-solving methodology that produces a solution to a problem in terms of self-contained entities called *objects*

Object

A thing or entity that makes sense within the context of the problem

For example, a *student*, a *car*, *time*, *date*

World View of OOD

Problems are solved by isolating the objects in a problem,

- determining their properties and actions (responsibilities), and
- letting the objects collaborate to solve a problem

An analogy: You and your friend fix dinner
Objects: you, friend, dinner
Class: you and friend are people
People have name, eye color, …
People can shop, cook, …
Instance of a class: you and friend are instances of class People, you each have your own name and eye color, you each can shop and cook
You collaborate to fix dinner
Class (or object class)
A description of a *group* of similar objects
Object (instance of a class)
A concrete example of the class
Classes contain fields that represent the properties (name, eye color) and behaviors (responsibilities) (shop, cook) of the class
Method
A named algorithm that defines behavior (shop, cook)

Top-Down Design
decomposes problems into **tasks**
Object-Oriented Design

decomposes problems
into collaborating
objects Steps

1. **isolate** the real-world objects in the problem
2. **abstract** the objects with like properties into groups (classes)
3. **determine** the responsibilities of the group in interacting with other groups

Object-Oriented Design Methodology

Four stages to the decomposition process

- Brainstorming to locate possible classes
- Filtering the classes to find duplicates or remove unnecessary ones
- Scenarios are tried to be sure we understand collaborations
- Responsibility algorithms are designed for all actions that classes must exhibit

Brainstorming

A group problem-solving technique that involves the spontaneous contribution of ideas from all members of the group

- All ideas are potential good ideas
- Think fast and furiously first, and ponder later
- A little humor can be a powerful force

Brainstorming is designed to produce a list of candidate classes

Filtering

Determine which are the core classes in the problem solution

There may be two classes in the list that have many common attributes and behaviors There may be classes that really don't belong in the problem solution

Scenarios

Assign responsibilities to each class

There are two types of responsibilities

- What a class must know about itself (**knowledge** responsibilities)
- What a class must be able to do (**behavior** responsibilities)

Encapsulation

The bundling of data and actions in such a way that the logical properties of the data and actions are separated from the implementation details

Each class **encapsulates** its data but shares their values through knowledge responsibilities

Responsibility Algorithms

The algorithms must be written for the responsibilities

- Knowledge responsibilities usually just return the contents of one of an object's variables
- Action responsibilities are a little more complicated, often involving calculations

Translation Process

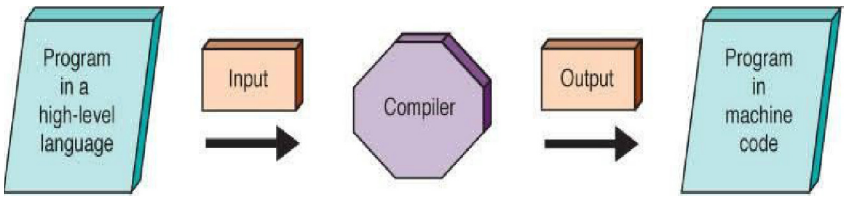
A program written in a high-level language must be translated into machine code
The machine code is then executed
Compilers and Interpreters are software tools employed to help with the translation process

Compilers

High-level language

A language that provides a richer (more English-like) set of instructions **Compiler**

A program that translates a high-level language program into machine code



Interpreter

A translating program that translates and executes the statements in sequence

- Assembler or compiler produce machine code as output, which is then executed in a separate step
- An interpreter translates a statement and then immediately executes the statement
- Interpreters can be viewed as *simulators*

Java

- Introduced in 1996 and became instantly popular
- Portability was of primary importance
- Java is compiled into a standard machine language called Bytecode
- A software interpreter called the JVM (Java Virtual Machine) takes the Bytecode program and executes it

Portability

The ability of a program to be run on different machines

Compiler portability

A program in a standardized language can be compiled and run on any machine that has the appropriate compiler

Bytecode portability

A program translated into Bytecode can be run on any machine that has a JVM

Programming Language Paradigms

Imperative Paradigm

Program describes the processing

Declarative Paradigm

Program describes the results

Each of these major paradigms have distinct subparadigms

Imperative

- Procedural
- Characterized by sequential instructions
- A program in which statements are grouped into a hierarchy of subprograms
- Fortran, C, C++
- Object-oriented model
- Program consists of a set of objects and the interactions among the objects • Python, Java, Smalltalk, Simula

Functionality of Imperative Languages

Sequence

Executing statements in sequence until an instruction is encountered that changes this sequencing **Selection**

Deciding which action to take

Iteration (looping)

Repeating an action

Strong Typing

Data type

A description of the set of values and the basic set of operations that can be applied to values of the type

Strong typing

The requirement that only a value of the proper type can be stored into a variable

Integer numbers
Real numbers
Characters
Boolean values
Strings

Declarations

Declaration

A statement that associates an identifier with a variable, an action, or some other entity within the language that can be given a name; the programmer can refer to that item by name **Reserved word**

A word in a language that has special meaning

Case-sensitive

Uppercase and lowercase letters are considered the same

Assignment statement

An action statement (not a declaration) that says to evaluate the expression on the right-hand side of
the symbol and store that value into the place named on the left-hand side

Named constant

A location in memory, referenced by an identifier, that contains a data value that cannot be changed

Input/Output Structures

Pseudocode algorithms used the expressions *Read or Get* and *Write or Print*
High-level languages view input data as a stream of characters divided into lines
Key to the processing
The data type determines how characters are to be converted to a bit pattern (input) and how a bit pattern is to be converted to characters (output)

name is a
string; age
is an
integer;
hourlyWage
is a real
The data must be a string,
an integer, and a real in
that order.

Control structures
An instruction that determines the order in which other instructions in a program are executed

Subprogram Statements

We can give a section of code a name and use that name as a statement in another part of the program When the name is encountered, the processing in the other part of the program halts while the named code is executed

Asynchronous processing

Not synchronized with the program's action

- *Clicking* has become a major form of input to the computer
- Mouse clicking is not within the sequence of the program
- A user can click a mouse at any time during the execution of a program

Functionality of OOPs

Encapsulation

A language feature that enforces information hiding **Classes**

Different meanings in different places (See next slide)

Inheritance

A property that allows a class to inherit the data and actions of another class

Polymorphism
An ability to handle the ambiguity of duplicate names
Object class (problem-solving phase)
An entity or thing that is relevant in the context of a problem
Object class (class) (problem-solving phase)
A description of a group of objects with similar properties and behaviors
Class (implementation phase) A pattern for an object
Object (implementation phase) An instance of a class

Class Definition
class encapsulates both data and actions

Inheritance and Polymorphism
Inheritance
A construct that fosters reuse by allowing an application to take an already-tested class and derive a class from it that inherits the properties the application needs
Polymorphism
The ability of a language to have duplicate method names in an inheritance hierarchy and to apply the method that is appropriate for the object to which the method is applied

Inheritance and
polymorphism work together
How?

They combine to allow the programmer to build useful hierarchies of classes that can be put into a library to be reused in different applications

Top-Down vs OO Designs
Top-down Solution

Data structures needed in solution are determined

Subprograms are written to manipulate the the data structures

Main program declares data structure

Main program calls to the subprograms, passing data structures as parameters

Object-oriented Solution

ADTs needed in solution are determined

ADTs are written only if not in library

Data structure is encapsulated within the class that implements the ADT

Main program is instructions to ADTs to perform the necessary tasks

Chapter 14

What Is Simulation?

Simulation

A model of a complex system and the experimental manipulation of the model to observe the results Systems that are best suited to being simulated are **dynamic**, **interactive**, and **complicated Model**

An abstraction of a real system

It is a representation of the objects within the system and the rules that govern the interactions of the objects

Continuous simulation

- Treats time as continuous
- Expresses changes in terms of a set of differential equations that reflect the relationships among the set of characteristics
- Meteorological models fall into this category

Queuing system

A discrete-event model that uses **random numbers** to represent the arrival and duration of events The system is made up of

- servers

- queues of objects to be served

To construct a queuing model, we must know

- The number of events and how they affect the system in order to determine the rules of entity interaction
- The number of servers
- The distribution of arrival times in order to determine if an entity enters the system
- The expected service time in order to determine the duration of an event

Meteorological models

Models based on the time-dependent partial differential equations of fluid mechanics and thermodynamics

Initial values for the variables are entered from observation, and the equations are solved to define the values of the variables at some later time

Computer models are designed to **aid** the weathercaster, not replace him or her

- The outputs from the computer models are predictions of the values of variables in the future
- It is up to the weathercaster to determine what the values **mean**

Relocatable models

Models applied to a moving target

Computational Biology
An interdisciplinary field that applies techniques of computer science, applied mathematics, and statistics to problems in biology
Encompasses bioinformatics, computational biomodeling, computational genomics, molecular modeling, and protein structure prediction.

Graphics
Originally the language of communications for engineers, designers, and architects
Computer-aided design (CAD)
A system that uses computers with advanced graphics hardware and software to create precision drawings or technical illustrations

Shape and surface influence an object's appearance
Equations used to describe planes, spheres, and cylinders
Real world surfaces are rough, which scatter light differently, requiring texture mapping techniques

Illumination model
Simulation of light interaction at one point on an object
Shading model (shading)
Process of using an illumination model to determine the appearance of an entire object **Rendering**
The process of creating an entire image

Computer Gaming

Computer gaming is a simulation of a virtual world

Game designers must have knowledge of the following to make people, objects, and environments behave realistically in a virtual world:

- Computer graphics
- Artificial intelligence
- Human-computer interactions and simulation
- Software engineering
- Computer security
- Fundamentals of mathematics
- Laws of physics relating to gravity, elasticity, light, and sound

Gameplay

Gameplay: The type of interactions and experiences a player has during the game Game genres, based on **gameplay,** include:

- **Action games**
- **Shooter games**
- **Action-adventure games**
- **Life-simulation games**
- **Role-playing games**
- **Strategy games**

Creating the Virtual World

Game engine--a software system within which games can be created Following functionality provided by tools of a game engine:

- A rendering engine for graphics
- A physics engine to provide a collision detection system and dynamics simulation
- A sound-generating component

Additional functionality resulting from tools of a game engine:

- A scripting language apart from the code driving the game
- Animation
- Artificial intelligence algorithms (e.g., path-finding algorithms)
- A scene graph that holds the spatial representation in a graphical sense **Soft Skills**

High Quality Game Design and Development Requires Effective Use of “Soft Skills”:

- Effective collaboration with designers, programmers, and artists on various technical ideas throughout the entire game design and development process
- Flexibility and adaptability as the game design constantly evolves and changes throughout the development and production process
- Willingness to abandon much of the completed design work when the game’s story line, mechanics, art, programming, audio, video, and/or scripting requires significant changes

Game Programming

After all the design decisions have been finalized, programmers produce the code to create the virtual world of the game
Popular languages include: C++, Java, and C
Some well-established game engineers have created custom languages based on their games, e.g., Epic Game's UnrealScript for the Unreal Game
A variety of application programming interfaces (APIs) and libraries are available to help developers with key programming tasks
The choice of API determines which vocabulary and calling conventions the programmer should employ to use the services
The target game platform determines which service the programmer will use; some libraries permit efficient cross-platform development
Coding process begins with the creation of "the game loop"
Game loop is responsible for managing the game world, regardless of any input from the user For example, the game loop might update enemy movement in the game or check for victory/loss conditions
Basically, the game loop manages the simulation
Large design teams have different programmers focus on different aspects of the game Thus, you might find yourself working as:
- A Junior Engine programmer writing and maintaining code for the game loop

- A 3D Software Programmer designing and implementing the 3D graphics component
- A User-Interface Programmer working on the APIs in the game engine
- Collaborateing effectively is essential to creating a streamlined, “killer” computer game
- Collaboration and cooperation will be necessary because, despite beta testing and demoing, invariably bugs may begin to adversely effect the operation of the game
- If it is an online game you helped develop, you will be able to perform any fixes without having to interrupt the ongoing action or force the company to order a costly recall of the game