Chapter 10

Streams and File I/O

- 10.1 Overview of Streams and File I/O
- 10.2 Text File I/O
- 10.3 Techniques for Any File
- 10.4 Basic Binary File I/O
- 10.5 Binary File I/O with Objects and Arrays
- 10.6 Graphics supplement

Objectives

- 1) Become familiar with the concept of an I/O stream
- 2) Understand the difference between binary files and text files
- 3) Learn how to save data in a file, using a Java program
- 4) Learn how to read data from a file, using a Java Program
- 5) Learn how to use the classes ObjectOutputStream and ObjectInputStream to write and read, respectively, class objects with binary files.

10.1 AN Overview of Streams and File I/O : I/O Overview

- *I/O* = Input/Output
  - » input to and output from programs
  - » Input : from keyboard or a file
  - » Output : to display (screen) or a file
- Advantages of file I/O
  - » permanent copy
  - » output from one program can be input to another
  - » input can be automated (rather than entered manually)

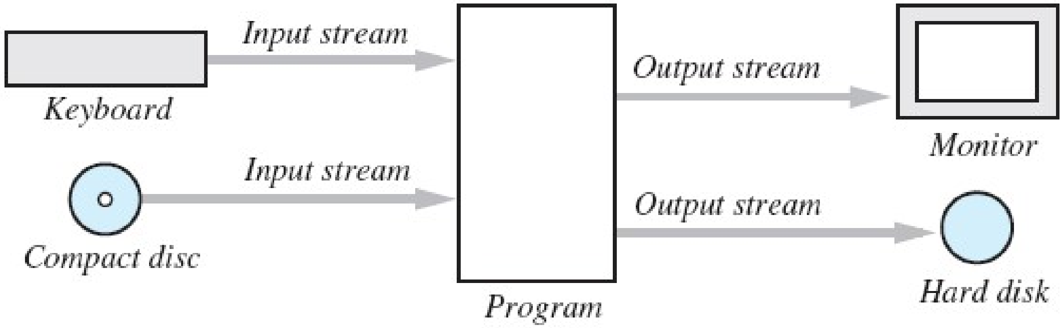
Note: Since the sections on text file I/O and binary file I/O have some similar information, some duplicate (or nearly duplicate) slides are included.

Streams

- ***Stream***: an object that either delivers data to its destination (screen, file, etc.) or that takes data from a source (keyboard, file, etc.)
  - » it acts as a buffer between the data source and destination
- ***Input stream***: a stream that provides input to a program
  - Scanner is an input stream
- ***Output stream***: a stream that accepts output from a program
  - » `System.out` is an output stream
- A stream connects a program to an I/O object
  - » `System.out` connects a program to the screen
  - » `Scanner` connects a program to the keyboard

The Concept of a Stream

- Streams are implemented as objects of special stream classes
  - » Class `Scanner`
  - » Object `System.out`
- Figure 10.1 I/O Streams



Binary Versus Text Files

- *All* data and programs are ultimately just zeros and ones
  - » each digit can have one of two values, hence *binary*
  - » *bit* is one binary digit
  - » *byte* is a group of eight bits
- ***Text files***: the bits represent
  - » one byte per character for ASCII, the most common code
  - » for example, Java source files are text files
  - » so is any file created with a "text editor"
- ***Binary files***: the bits represent other types of encoded information, such as executable instructions or numeric data
  - » these files are easily read by the computer but not humans
  - » they are
    - – actually, you *can* print them, but they will be unintelligible
    - – "printable" means "easily readable by humans when printed"

끝