Example: Thinking about objects

- Think about a book:
  - what are its features? pages, title, author, content, ...
  - what actions can you take on or with a book? read, write, turn page, ...
- A book is a general abstraction: it denotes a generic type
- The course textbook is a specific instance of a book

Example: Thinking about objects (cont)

- There are different types of books as well:
  - sci fi, mystery, fiction, non-fiction, historical, romance, ...
  - each of these types is also a general abstraction
- specific titles are specific instances
  - e.g., War and Peace, Secrets and Lies, An Introduction to Object-oriented Programming with Java, ...

Exercise: Thinking about objects

- Consider a system for organizing all of your media files on your computer. How would you approach this problem?
Object-oriented programming (OOP)

“A way of thinking of computer programs as a group of objects that work together, each object containing everything it needs to handle a specific task.”


Back to the “book” example

- Book: generic type
  - class
- Fiction book: a particular type of book, but still generic
  - subclass
- War and Peace: specific instance of a book
  - object
- Number of pages, title, author, ...: features of a book
  - variables
- Read, write, turn page: actions you can take on a book
  - methods

Definitions

- **Class**: definition or template of an object
  - class Book
- **Object**: specific instance of a class
  - Book book1;
- **Method**: action that a class performs
  - book1.turnPage(32);
- **Variable**: data used by a class (also called data member)
  - book1.title = “War and Peace”;

Exercise

- Go back to your media organizer plan. Point out examples of classes, variables, and methods.
- What would be an example of an object in your system?
Procedural programming

- Another style of programming
- Focus is on mechanism for completing a certain task
- “Sequential”, “linear”, not modular

Difference between OOP and procedural programming

“Historically, programs have been viewed as procedures (or we may think of these as "verbs") that operate on data. OOP takes the view that programs should start by thinking about the data (or "nouns") first.”

- from http://www.pdacortex.com/glossary.htm

Types of languages, part 2

- Compiled
  - text --> machine code --> execution
  - run program: follow machine code instructions
  - examples: C, C++, Fortran
- Interpreted
  - text --> execution
  - run program: read each line and interpret on the fly
  - examples: Perl, Python, PHP, Javascript, Mathematica

Q: To what category does Java belong?
What happens when we write a Java program?

- Step 1: Write the program in a text editor
  - saved as human-readable text
- Step 2: **compile**
  - converts text into numerical code called **bytecode**
  - **bytecode** is a language that's understood by the Java interpreter
- Step 3: run the program
  - program is **interpreted**

A: Java is both compiled and interpreted!

- Bytecode is a machine-like code that's understood by many different platforms
- In order to execute a Java program, you need an **interpreter** to read the bytecode and execute the instructions based on whatever system the code is being run on
  - in other words, the same instruction will be carried out slightly differently on Mac, Windows, and Linux
  - the interpreter program is called the **Java Virtual Machine (JVM)**