

Chapter 2

Object Oriented Design

Object Oriented Design

- ◆ Combining data and the operations on that data - creating an object
- ◆ Classes are an excellent example of an object

10/2/04

Object Oriented Design

2

Object-Oriented Design Goals

- ◆ Robustness
 - Handle unexpected input
 - ALWAYS error check EVERYTHING
 - ◆ Buffer overrun errors - serious security issues
 - Do not exit



10/2/04

Object Oriented Design

3

Object-Oriented Design Goals

- ◆ Adaptability
 - Year 2000 issue
 - Different implementations of C++ allow you to implement features differently
 - ◆ -pedantic

10/2/04

Object Oriented Design

4

Object-Oriented Design Goals

◆ Reusability

- Do not reinvent the wheel
 - ◆ Libraries, templates, STL
- In this course, you will be writing the reusable code
 - ◆ You must learn **how** to code first
 - ◆ Learn to add, subtract, multiply, etc before you can use the calculator

10/2/04

Object Oriented Design

5

Object-Oriented Design Goals

◆ Reusability

- Code with reusability in mind
- Never, ever, ever copy and paste code!
Find a general implementation and reuse

10/2/04

Object Oriented Design

6

Object-Oriented Design Principles

◆ Abstraction

- Define functions and not implementation
- Abstract Data Type (ADT)
 - ◆ Data and operations on that data
 - ◆ Operations are specified with a defined interface
 - ◆ Specify what and not how
- You will use classes to design ADTs this quarter

10/2/04

Object Oriented Design

7

Object-Oriented Design Principles

◆ Encapsulation

- Information hiding
 - ◆ Classes and functions
- User does not need to know how something is done, just that it is done.
- Objects encapsulate data and operations
- Functions encapsulate actions

10/2/04

Object Oriented Design

8

Object-Oriented Design Principles

◆ Encapsulation



Honda Civic - Is this the gas or the gas/electric version?

Object-Oriented Design Principles

◆ Modularity

- Design program as interacting pieces or modules
- Closely related to reusability
 - ◆ Design a module that you can reuse later

Object-Oriented Design Principles

◆ Modularity

- Video game design
 - ◆ Sell graphics engine



Benefits of Modularity

◆ Program construction

- Team work
- Easier to manage

◆ Debugging

- Isolates errors
- Debug each module as you go
 - ◆ Easy to locate bugs
- Do not write entire program and then debug
 - ◆ Difficult to locate bugs

Benefits of Modularity

- ◆ Reading the program
 - Easier to read
 - Helps designer construct complicated project just as it helps a reader understand
 - Functions
 - ◆ Easy to read if they have a good description and descriptive names and parameters
 - ◆ Reader should not have to actually read the code in the function unless they require details

10/2/04

Object Oriented Design

13

Benefits of Modularity

- ◆ Modifying the program
 - Isolates modifications
 - Update just a module as it is changed
 - ◆ Game and application patches
- ◆ Eliminating redundant code

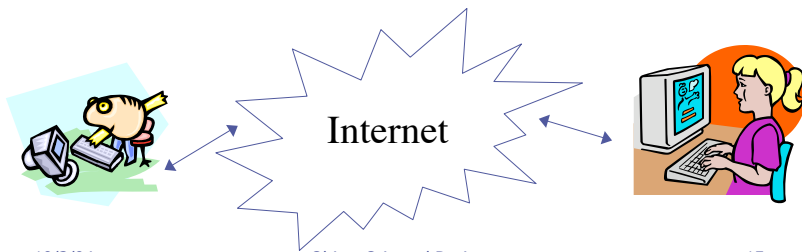
10/2/04

Object Oriented Design

14

Black Box Design

- ◆ Defined inputs and outputs
- ◆ Defined actions
- ◆ User knows nothing of internal workings



10/2/04

Object Oriented Design

15

Black Box Design

- ◆ In class exercise - give black box design of ATM

Inputs



Outputs



10/2/04

Object Oriented Design

16

Programming Assignments

- ◆ I will give you detailed specifications
 - Skeleton code
 - Detailed function interface
 - Detailed output
 - ◆ Must match EXACTLY
- ◆ You will write the implementation