In Class Demonstration

• War, the card game
• A unique variant: ties decided by suit
• Output:
  • Each turn, the cards that were played
  • End of the game: who won and how many turns the game lasted
Stack vs. Heap Memory

• Stack variables manage memory on their own
• Order of construction and destruction of multiple objects
  • LIFO – Last In First Out
• Heap memory – you have to manage it yourself
• Example in Visual Studio
  • Memory must be deallocated
Classes

• Classes define types for new objects and methods they share
  • Objects sharing a class are called members of that class
• Functions attached to object types are member functions a.k.a methods
• Constructors: methods run after the object is first allocated
• Other member functions: Inherited by all members of the class
Public/Private Member Functions

• Public – can be called from anywhere
• Private – can only be called from within the object
• Security: Assign the minimum permissions possible
  • Avoid public object variables – use getters and setters instead
  • If a method does not need to be public, make it private
• Always code like someone else will inherit your code
  • And that person knows nothing about your program and how it works
    • That person could be you
  • Or, that person is evil and intentionally tries to screw things up
Initializer Lists

• Demonstration in Visual Studio

• More efficient than initialization in the body
  • Why? Construction of empty variable, then overwritten

• Also situations in which you must use initializer lists
  • Field of your object has no default constructor
    • Parent of object has no default constructor
    • Or, you don’t want to use the default constructor
  • Const fields or references; ensures initialized exactly once

• Clarity vs. correctness
Const Correctness

• Const member functions: cannot change object
  • Const objects can only call const functions
  • Non-const objects can call both const and non-const functions
  • When possible, declare as const

• Const function arguments – cannot be changed by function
  • Mostly matters if passing references, because of scope

• Const references as arguments
  • Good for keeping function call overhead down