

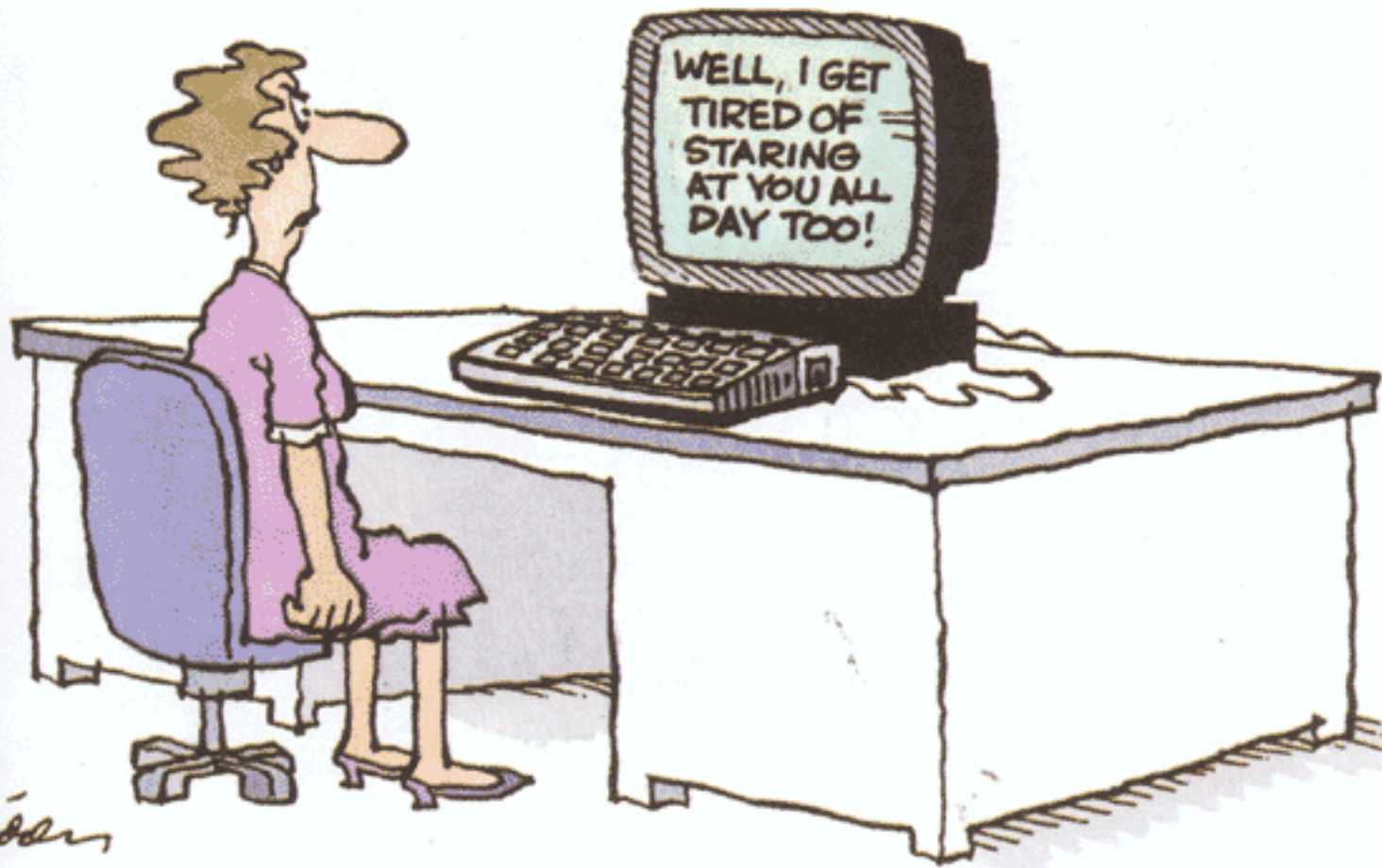
# C++

SWE315

Lesson 8

Prof. Zachi Baharav

[zbaharav@cogswell.edu](mailto:zbaharav@cogswell.edu)



Fiddler

# Lesson 8

- In this lesson:
  - Deep / Shallow copy.
  - Copy constructor and Assignment.
  - Static
  - Lab: work
  - Good easy resource:
    - <http://www.learncpp.com/cpp-tutorial/911-the-copy-constructor-and-overloading-the-assignment-operator/>
    - <http://www.learncpp.com/cpp-tutorial/912-shallow-vs-deep-copying/>

# Static

- Class member:
  - One copy for all the objects!
  - Initialized once in the .cpp file. (unless const, in which case can be in header)
  - Usages: Flag, counter.
- Member function:
  - Invoked with no object: Doesn't get a \*this pointer!
    - Simply invoked with class-name and scope-resolution.
  - Can only access static data members (because it doesn't have any of the other object elements)
  - Usages: e.g., setting a static flag used by all the objects (formatting etc).

# Implicit member functions

- Default constructor
- Copy constructor ←
- Assignment operator ←
- Default destructor
- Address operator

# Copy .vs. Assignment

- Copy : Into NEW object.
- Assignment: Into existing object.
  
- There are three general cases where the copy constructor is called instead of the assignment operator:
  - When **instantiating one object and initializing it** with values from another object.
    - Point P1(0,0) ;
    - Point P2 = P1 ; // <-- COPY constructor
  - When **passing an object by value.**
  - When an object is **returned from a function by value.**

# Summary (Copy/assignment)

- The default copy constructor and default assignment operators do shallow copies, which is fine for classes that contain no dynamically allocated variables.
- Classes with dynamically allocated variables need to have a copy constructor and assignment operator that do a deep copy.
- The assignment operator is usually implemented using the same code as the copy constructor, but it :
  1. checks for self-assignment,
  2. returns \*this, and
  3. deallocates any previously allocated memory before deep copying.
- If you don't want a class to be copyable, use a private copy constructor and assignment operator prototype in the class header.

END