

# 15-213

*“The course that gives CMU its Zip!”*

## **Dynamic Memory Allocation I**

### **October 28, 2003**

#### **Topics**

- **Simple explicit allocators**
  - **Data structures**
  - **Mechanisms**
  - **Policies**

# Harsh Reality

## *Memory Matters*

### Memory is not unbounded

- It must be allocated and managed
- Many applications are memory dominated
  - Especially those based on complex, graph algorithms

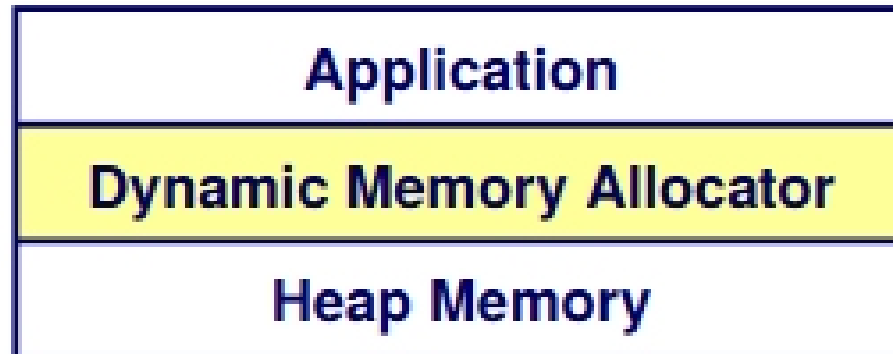
### Memory referencing bugs especially pernicious

- Effects are distant in both time and space

### Memory performance is not uniform

- Cache and virtual memory effects can greatly affect program performance
- Adapting program to characteristics of memory system can lead to major speed improvements

# Dynamic Memory Allocation



## Explicit vs. Implicit Memory Allocator

- **Explicit:** application allocates and frees space
  - E.g., `malloc` and `free` in C
- **Implicit:** application allocates, but does not free space
  - E.g. garbage collection in Java, ML or Lisp

## Allocation

- In both cases the memory allocator provides an abstraction of memory as a set of blocks
- Doles out free memory blocks to application

**Will discuss simple explicit memory allocation today**