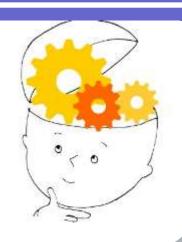
# Lesson 5: Memory Management

Areas of running program

Managing dynamic memory



# 5.1 The 5 Memory Areas

- 5 areas of a running C++ program
  - Const data
  - Stack
  - Free store
  - Heap
  - Global / static
- Guidelines

### 5.2 Header File <new>

- Provides features to manage memory
  - Functions
  - Types
  - Constants
- Additional notes

### 5.3 Memory Exhaustion – bad\_alloc

- Sample code
  - Demonstrates running out of memory
  - bad\_alloc thrown

## 5.4 Memory Exhaustion - nothrow

- Sample code
  - Demonstrates running out of memory
  - bad\_alloc *not* thrown

### 5.5 Registering Memory Allocation Handler

- Register handler called when out of memory
- Uses:
  - Can free up memory to allow app to keep running

#### 5.6 Call Global Allocators & Deallocators

- Call global memory allocators
- Call global memory deallocators

#### 5.7 Replace Global Allocators & Deallocators

- Replace global memory allocators
- Replace global memory deallocators
- Benefit
  - Can control global memory management
- Dangerous

#### 5.8 Custom Class Allocators & Deallocators

- Replace class' memory allocators
- Replace class' memory deallocators
- Benefit
  - Control when expensive objects created / destroyed
  - Implement memory pool
- Guidelines

### 5.9 Quiz: What's Wrong with Code?

- Analyze code
  - Determine how allocators & deallocators are being used incorrectly
- Guidelines

### 5.10 Quiz: Which Deallactors Get Called?

- Analyze code
  - Understand behavior with arrays
- Guidelines

## 5.11 auto\_ptr

- auto\_ptr
  - Standard class that cleans up dynamically allocated object
- Sample unsafe code not using auto\_ptr
- Sample safe code using auto\_ptr
- Resource Acquisition Is Initialization (RAII)
- Guidelines