

# Dynamic Variables

- sometimes you won't know how many variables you'll need until your program runs
- in this case, *dynamically* ask the OS to reserve new memory for variables
- OS will allocate room for your variable in the free memory, then returns the address of the new variable
- when you're done with the variable, you can tell the OS to free the space

## New and Delete

- **new** command requires two pieces of info
  - what type of array you want to allocate
  - how many slots you want in your array
- when you're done, use **delete** to free the array

## Using new and delete in Classes

- dynamically allocate (typically) using **new** in constructor or designated function
- deallocate (typically) using **delete** in destructor

## Dynamic Allocation of Non-Array Variables

- calling delete on **nullptr** won't crash; delete command is smart enough to prevent that

```
1 int main() {
2     int *arr
3     int size;
4
5     cin >> size;
6
7     arr = new int[size]; // Dynamically allocates new integer array of size "size"
8
9     arr[0] = 10;
10    arr[2] = 75;
11
12    delete[] arr; // Frees up space reserved for integer array
13 }
```