Compiler-Generated Members

- If a class doesn't provide them, the compiler will automatically generate the following members if it needs them
  - Default (no-arg) constructor
  - Copy constructor (member-wise initialization)
  - operator = (member-wise assignment)
  - Destructor
When are constructors and destructors called?

- Static variables
- Local variables
- Parameters
- Return values
- Heap objects
- Array elements
- Temporary objects
- Class objects as members
- Inheritance
When are constructors and destructors called?

- Static variables
  - Global static and Class static variables are constructed before main begins and destructed after main exits
When are constructors and destructors called?

- **Local variables**
  - Constructed each time the program passes through their declaration
  - Destructed each time the variable's block is exited

```cpp
for (int x=0; x < 10; ++x) {
    string message("Greetings, earthling");
    cout << message << endl;
}
```
When are constructors and destructors called?

- Parameters
  - Object parameters that are passed by value
  - When the function is called, the parameter is constructed with the actual parameter value
  - Destructed when the function completes

```c++
void PrintString(string s) {
    cout << s << endl;
}

void main() {
    PrintString("fred");
}
```
When are constructors and destructors called?

- **Return values**
  - Objects that are returned by value from a function
  - When the function returns, the return value on the stack is constructed with the value specified in the return statement
  - Destructed when the calling function is done with it

```cpp
string GetMessage() {
    string message = "Greetings, earthling";
    return message;
}

void main() {
    string msg = GetMessage();
    cout << msg << endl;
}
```
When are constructors and destructors called?

- **Heap objects**
  - Constructed when `new` is called
  - Destructed when `delete` is called

```cpp
void main() {
    string * msg = new string("Greetings, earthling");
    cout << *msg << endl;
    delete msg;
}
```
When are constructors and destructors called?

- **Array elements**
  - When an array is created, each element is constructed using its default (no-args) constructor
  - When an array is destroyed, each element is destructed

```cpp
void DoIt() {
    string someStrings[10];
    string * moreStrings = new string[20];
    ...
    delete [] moreStrings;
}

void main() {
    DoIt();
}
```
When are constructors and destructors called?

- Temporary objects
  - When evaluating expressions, the compiler sometimes needs to create temporary objects
    ```
    string msg = s1 + s2;
    // the result of s1 + s2 is stored in a temporary object before the assignment
    ```
  - Constructed when created during expression evaluation
  - Destructed after the expression has been evaluated (no later than the end of the block)
When are constructors and destructors called?

- Class objects as members
  - Classes can have member variables that are objects
  - A member object is constructed before the containing object's constructor is called
  - A member object is destructed after the containing object's destructor has finished
When are constructors and destructors called?

- Inheritance
  - When a subclass instance is created, the superclass constructor is executed before the subclass constructor
    - In an inheritance chain, constructors are executed from top to bottom
  - When a subclass instance is destroyed, the superclass destructor is executed after the subclass destructor
    - In an inheritance chain, destructors are executed from bottom to top