Intermediate/Advanced Computer Programming
Constructors

Place to *initialize* member data and to allocate other resources needed (e.g. memory, files, sockets)

When code in constructor body is called, data members are already allocated

Compiler will generate a default constructor if any constructor is not defined
Constructor

Before calling constructor

Object

data members:

x = 9827432
y = 28374243
Constructor

After calling constructor

Object

data members:

x = 0
y = 0
**const**

Informs compiler that a value/object should not be changed

Must initialize variable when the variable is defined

```cpp
const int x = 5;
//error: uninitialized const 'x'
const int x;
```

What about data members of an object marked as `const`?
class engine {
    const int cylinders;
};

class engine::engine()
{
    // error cylinders is const
    cylinders = 8;
}

const
Initialization List

Initialize objects/data as they are allocated

Foo:Foo() :Initialization List {}

double:

engine:engine(): cylinder(8){}
Composition

Classes allow the programmer to define user-defined data types

- User-defined data types allow the programmer to better describe real-world data

Just as objects can contain native data types, objects can also contain other objects
Composition

Clock object

Time object

int hour;
int minute;
int second;

Time class
IncrSeconds(...);
getTime(...);
GetTime(...);

Clock class
DisplayTime24(...);
DisplayTime12(...);
OOP

Encapsulation
Inheritance
Polymorphism
Inheritance

Allows one to **reuse** and extend existing classes

**Base class**
- existing class

**Derived class**
- new class that uses the base class's code

Allows different abstractions
- Is-a-kind-of
- has-a
Inheritance

Another type of client, another type of interface

**Public** interface
- For unrelated clients

**Protected** interface
- For clients that inherit from the class

**Private** interface
- Generally for members/friends of the class
Inheritance

Clock

Alarm Clock
Inheritance and Composition

Clock object
- Time object
  - int hour;
  - int minute;
  - int second;
  - bool display24;

Time class
- incrSeconds(...);
- getTime(...);
- setTime(...);
- displayTime24(...);
- displayTime12(...);

Clock class
- setCurrentTime(...);
- detDisplayType(...);
- display(...);

AlarmClock object
- Time object
  - int hour;
  - int minute;
  - int second;

AlarmClock class
- setAlarm(...);
- snooze(...);
- disable(...);
- enable(...);
- display(...);
Virtual Functions

Clock

display()

AlarmClock

display()
Virtual Functions

Clock c;
AlarmClock ac;

Clock *myClocks[2];
myClocks[0] = &c;
myClocks[1] = &ac;

myClocks[0].display();
myClocks[1].display();