Object-Oriented Programming
with C++

This course is about...
- Object-oriented programming and how it extends the power of structured programming;
- C++ and how it extends the power of C (classes, overloaded operators, templates, inheritance, polymorphism, exceptions);
- Event-driven interactive software.

Introduction

Object-oriented programming

Programming with
- Classes
- Encapsulation
- Inheritance
- Polymorphism

For
- Tools adapted to how we think
- Code reuse
- Safety

The right mouse button is “object oriented”

- Point to an object (icon, button, window, etc.) and click with right button.
- Operations (methods) are listed on the menu that appears.
- Data members are listed under “properties”. You can change certain properties of an object.

New paradigms

- Software organized around data types, not procedures;
- Instances of these types interact with their environments via messages. An interaction sequence, unlike an algorithm, has no finite time boundary;
- Intelligent agents are a kind of object that can only be implemented in such an object-oriented, interactive environment.

What is this?

```c
struct X
{
    widgets A[100];
    int num_widgets;
};
typedef struct X x;

void init(x & c);
void insert(x & c, widgets w);
```

What are more appropriate names or descriptions for X, x, c, w?
Elements of the C and C++ languages

C++ (mid-80s) is an extension of C (1971).

Object technology

- Analysis, design, programming, operating systems, databases
- An alternative to modular decomposition in system design
- Encapsulation
- Objects are instances of classes
- Inheritance: class hierarchies

A C++ class declaration

```cpp
struct employees
{
  char name[40];
  int salary,
  hours_worked;
  calculate_paycheck();
};
```

Four special kinds of class

- **Container** (collection): Class whose instances contain instances of another class
- **Iterator**: Iterator objects scan through collections
- **Derived class**: Class that inherits from another class
- **Base class**: Class from which derived classes may inherit

Inheritance in C++

- **Derived class** is a relationship between instances of classes
- **Base class** is a relationship between classes

```
struct instructors : employees
{
  char office[40];
};
```

- An instructor has all the properties and operations of an employee, plus an office.

Containment vs. inheritance
An object has...
- **Abstraction and typing**: It is an instance of a class
- **Encapsulation**: It is separate from other objects
- **Modularity**: Objects may contain other objects
- **Hierarchy**: Its class may be derived from other classes
- **Concurrency**: Multiple objects of the same class may exist at the same time
- **Persistence**: Most objects exist for a duration

Advantages of object technology
- Natural way to model real-world situations
- Uses familiar concepts
- Increases productivity in application development
- Facilitates code reusability

Topics
1. Using streams, other predefined classes
2. Classes and encapsulation
3. Operator overloading
4. Templates
5. Collections and iterators
6. Object-oriented design
7. Inheritance
8. Polymorphism
9. Exception handling