The Basics of Object-Oriented Programming

Object Oriented Programming

- Alternative to Procedural Programming - Design is centered around the Objects
- In Procedural Programming, the modules are procedures
- In Objet-Oriented programming, the modules are classes

Object Oriented Programming

- A *class* is a collection of objects
- Objects in a class share properties (attributes)
  - For example: the class of *Car is a collection of objects; they share the attributes of having 4 wheels, an engine, a model name, an exterior color, interior fabric, and baseprice*

A Note on Attributes

- An *Attribute* (or field in C++) is a property that is associated with an object.
- The attribute describes the object and holds some value required for processing

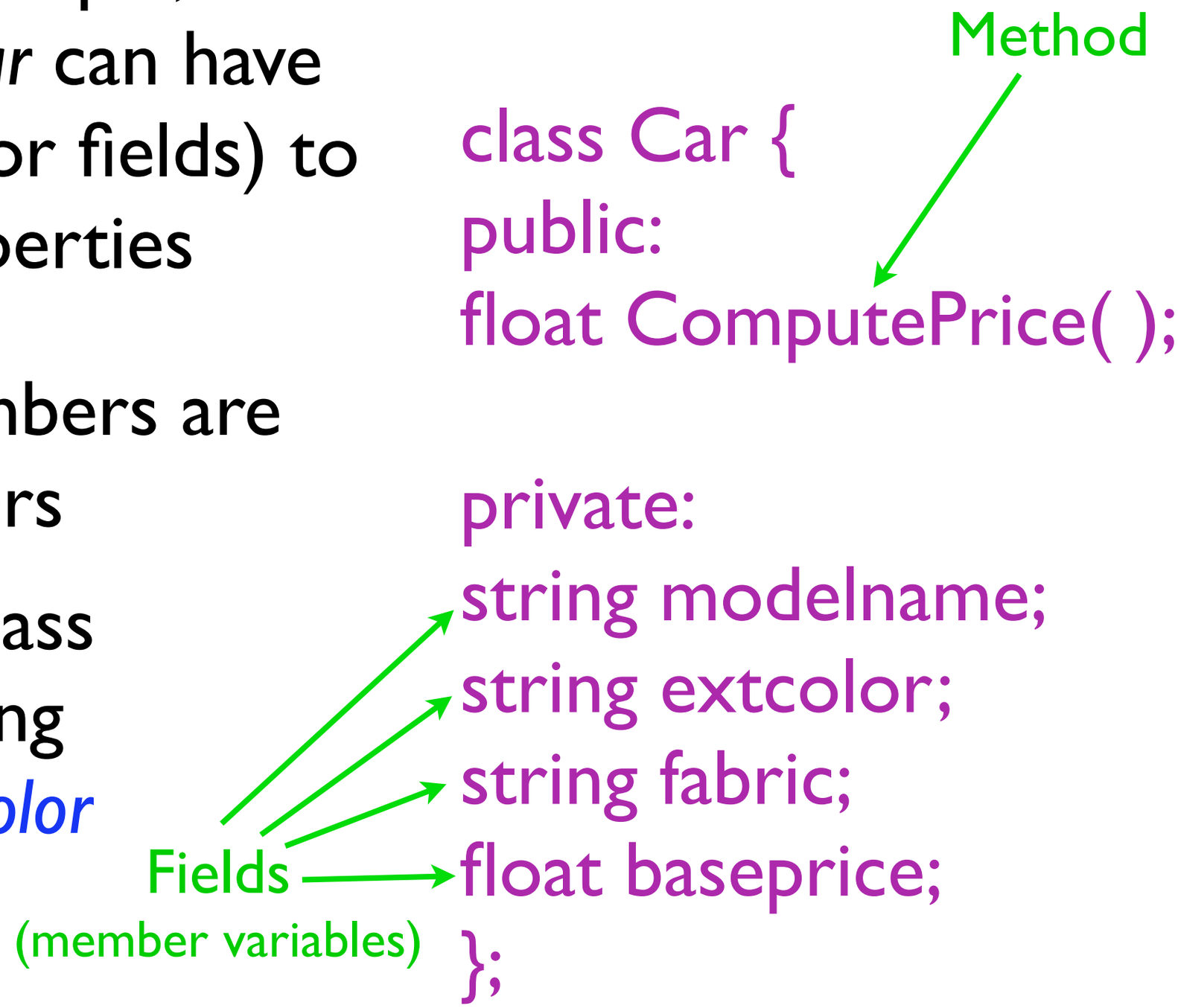
Object Oriented Programming

- In an object-oriented language, such as C++, a class is a data type.
  - variables are defined in the usual manner
  - for example, if Car is a user defined data type in C++, then we can define a variable such as *Ford* to represent the object.
  - That is, the Car *Ford* belongs to the class of Cars

In the previous example, a Variable of type *Car* can have *member variables* (or fields) to represent the properties

In C++, these members are called data members

For example, the class *Car* may have a string variable called *extcolor*



Constructing a Class

```
class Car {
public:
  float ComputePrice( ) {
      return baseprice;  }

private:
string modelname;
string extcolor;
string fabric;
float baseprice;
};
```

public
member of class

method defined within the class

private
members
(the data members)

Assignment

- Think of a student management system that a Registrar might use
- Write down the class name, any attributes (fields) that you think the class should have and any methods (functions) that should be included in the class
    - Note: this does not have to be done in C++, just a list.