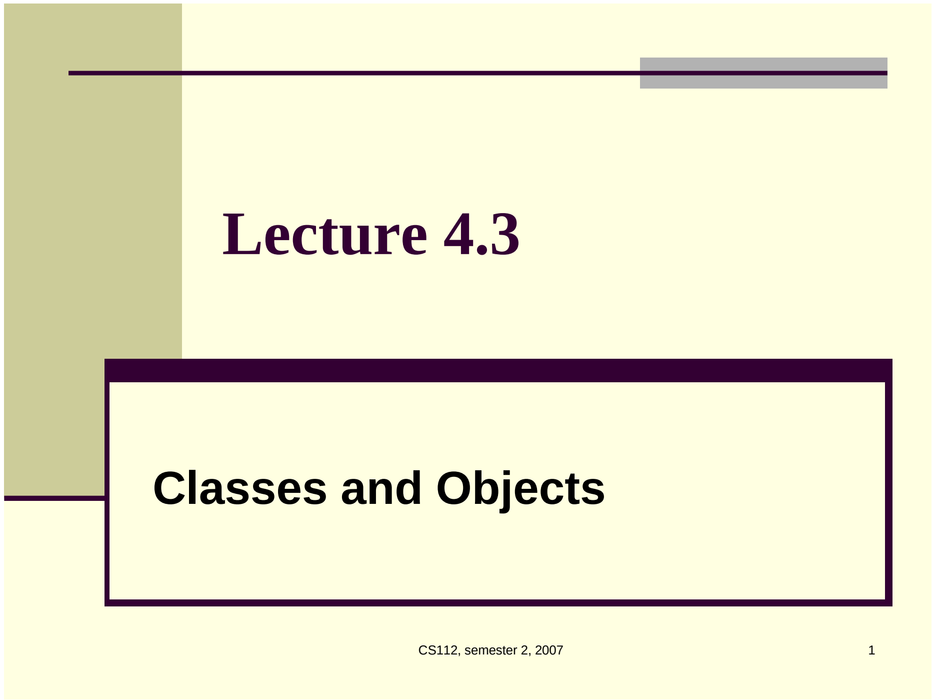
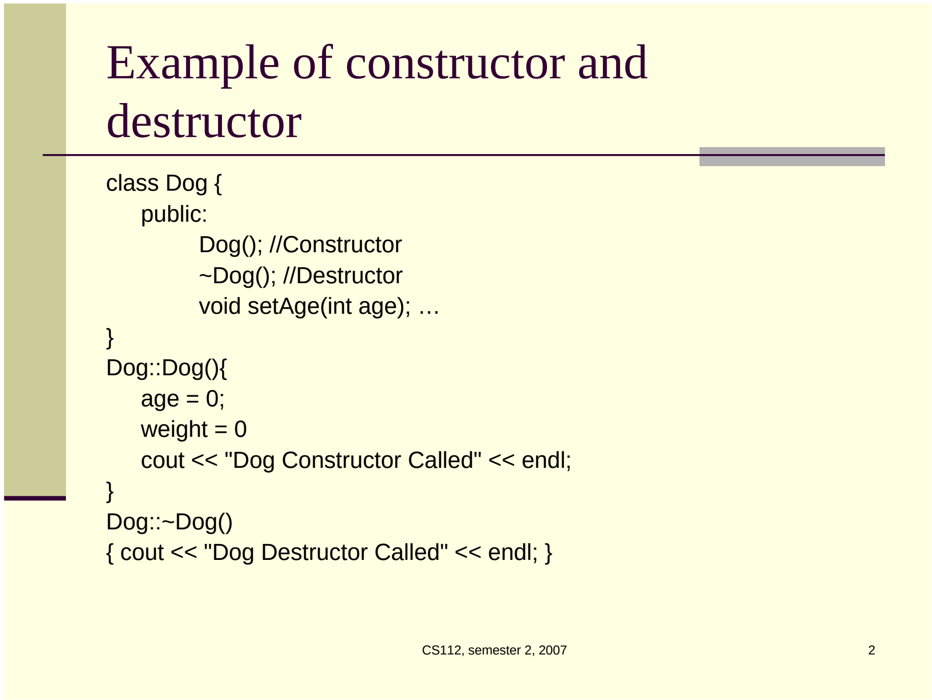


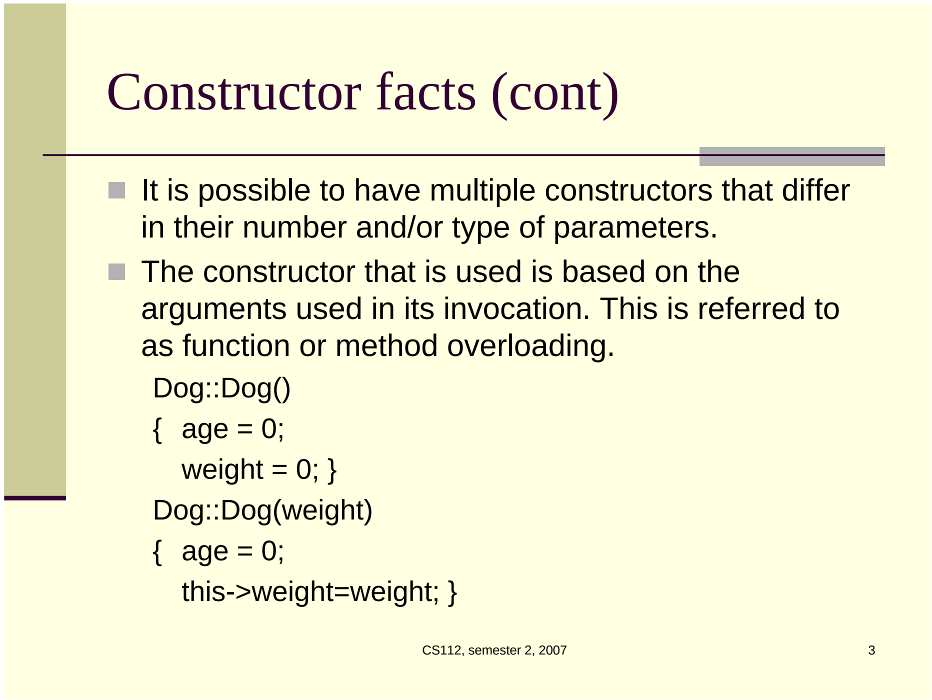
Lecture 4.3

Classes and Objects



Example of constructor and destructor

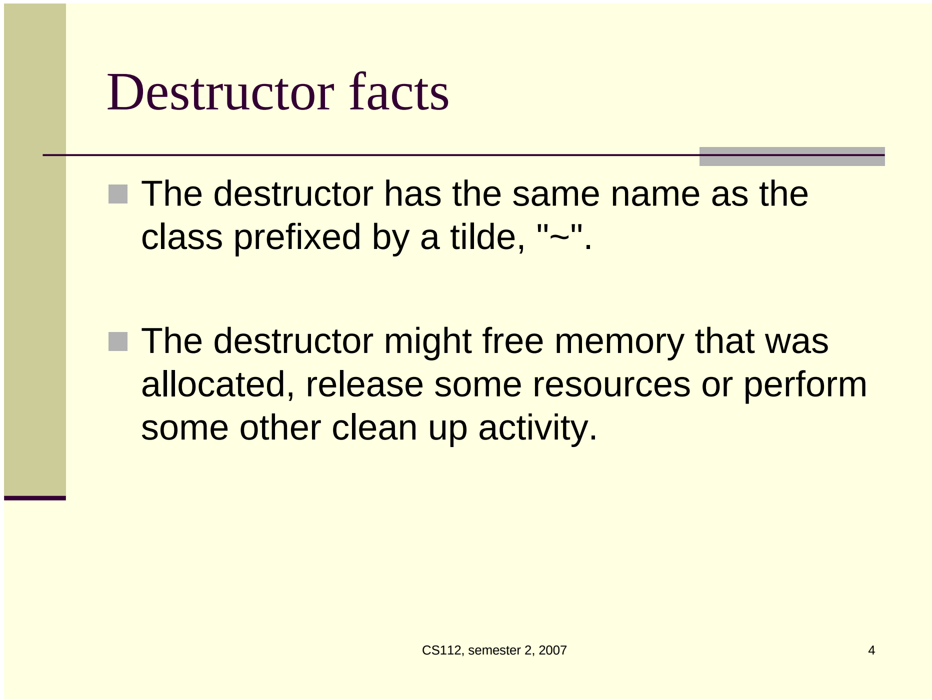
```
class Dog {
    public:
          Dog(); //Constructor
          ~Dog(); //Destructor
          void setAge(int age); …
}
Dog::Dog(){
    age = 0;
    weight = 0
    cout << "Dog Constructor Called" << endl;
}
Dog::~Dog()
{ cout << "Dog Destructor Called" << endl; }
```



Constructor facts (cont)

- It is possible to have multiple constructors that differ in their number and/or type of parameters.
- The constructor that is used is based on the arguments used in its invocation. This is referred to as function or method overloading.

```
Dog::Dog()
{  age = 0;
   weight = 0; }
Dog::Dog(weight)
{  age = 0;
   this->weight=weight; }
```



Destructor facts

- The destructor has the same name as the class prefixed by a tilde, "~".
- The destructor might free memory that was allocated, release some resources or perform some other clean up activity.

Using Objects

- The following program declares objects of the Dog class.
- For simplicity, all code will be contained in a single source file
- In larger projects classes are usually kept in separate files from the main program.

Using Objects Example

```
#include <iostream.h>

class Dog {
    private:
      int age;
      int weight;
```

Using Objects Example (cont)

```
public:
   Dog(); //Constructor
   ~Dog(); //Destructor
   void setAge(int age);
   int getAge();
   void setWeight(int weight);
   int getWeight();
   void speak();
};
```

Example (cont2)

```
Dog::Dog()
{
    age = 0;
    weight = 0;
    cout << "Dog Constructor Called" << endl;
}
Dog::~Dog()
{
   cout << "Dog Destructor Called" << endl;
}
```

Example (cont3)

```
void Dog::setAge(int age){
   this->age = age;}
int Dog::getAge(){
   return age;}
void Dog::setWeight(int weight){
   this->weight = weight; }
int Dog::getWeight(){
   return weight; }
void Dog::speak(){
   cout << "BARK!!" << endl; }
```

Example (cont4)

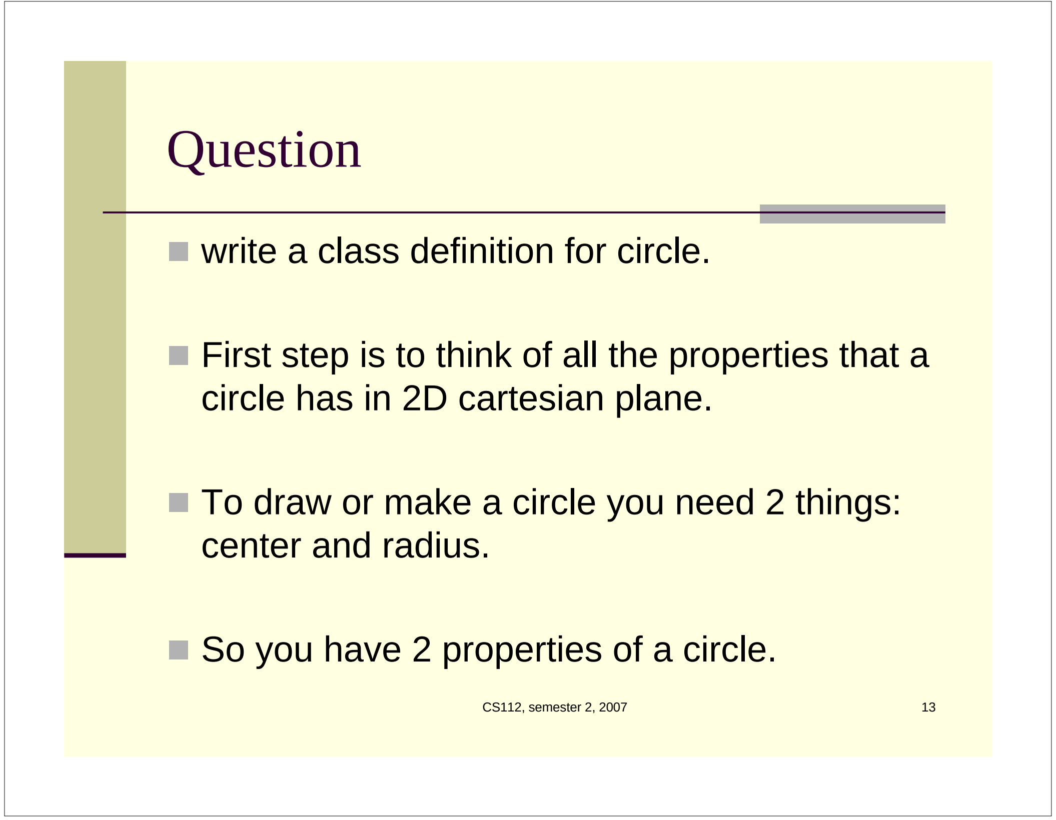
```
int main()
{
   Dog fido;
   Dog rover;
   cout << "Rover is " << rover.getAge() << " years old." ;
   cout << "He weighs " << rover.getWeight() << " lbs." ;
   cout << "Updating Rover's Age and Weight" << endl;
   rover.setAge(1);
   rover.setWeight(10);
```

Example (cont5)

```
cout << "Rover is " << rover.getAge() << " years old.";
cout << "He weighs " << rover.getWeight() << " lbs.";

cout << "Fido is " << fido.getAge() << " years old.";
cout << "He weighs " << fido.getWeight() << " lbs.";

cout << "Setting Fido to be the same as Rover" ;
fido = rover; //NOTE: Object assignments are not
                         //recommended
```

Example (cont6)

```
cout << "Fido is " << fido.getAge() << " years old." ;
cout << "He weighs " << fido.getWeight() << "lbs.";

rover.speak();
fido.speak();

return 0;
}
```



Question

- write a class definition for circle.
- First step is to think of all the properties that a circle has in 2D cartesian plane.
- To draw or make a circle you need 2 things: center and radius.
- So you have 2 properties of a circle.