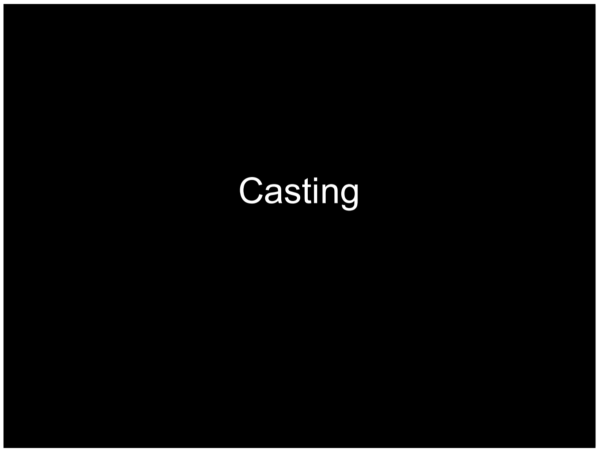
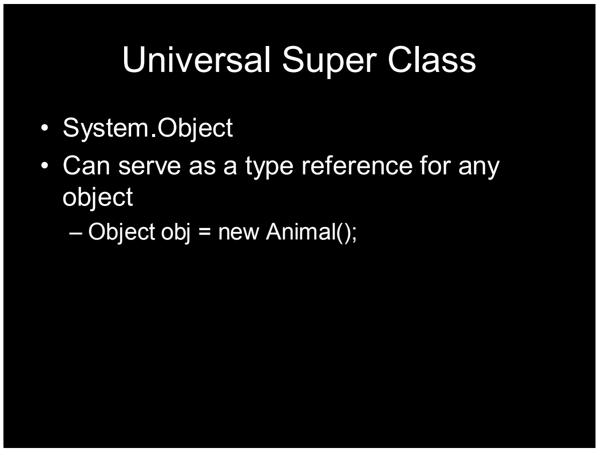
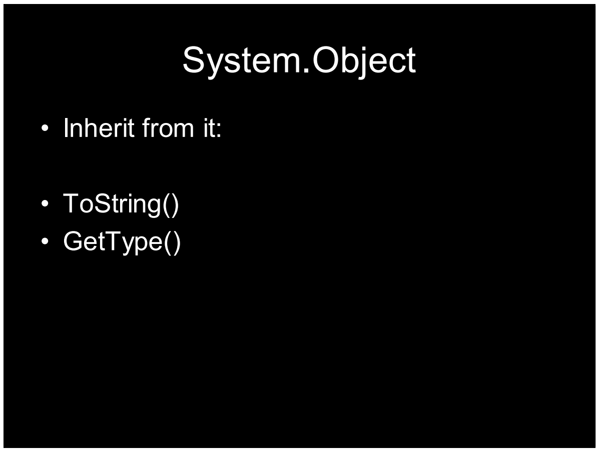


Casting



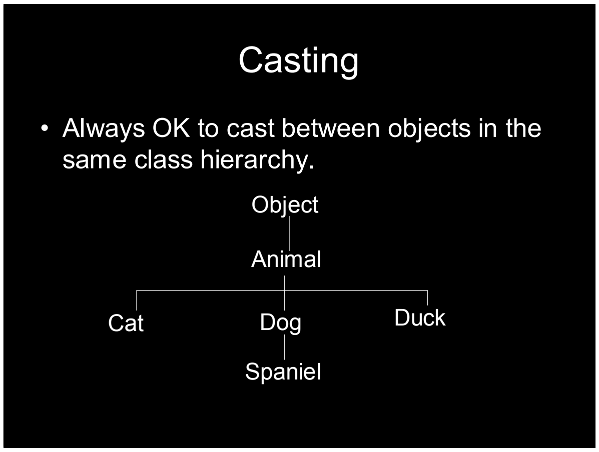
Universal Super Class

- System.Object
- Can serve as a type reference for any object
  - – Object obj = new Animal();



System.Object

- Inherit from it:
- ToString()
- GetType()



Casting

- Always OK to cast between objects in the same class hierarchy.

Object

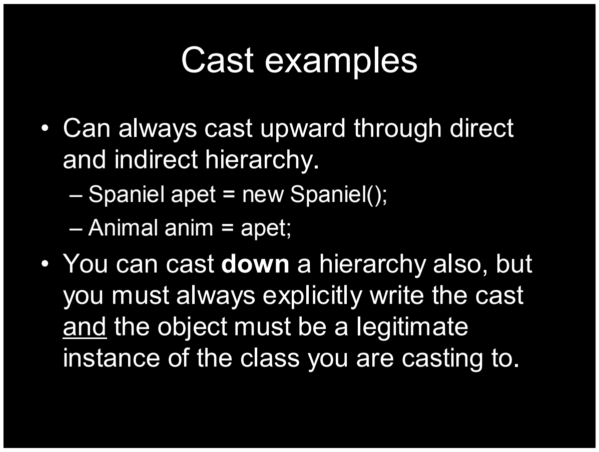
Animal

Cat

Dog

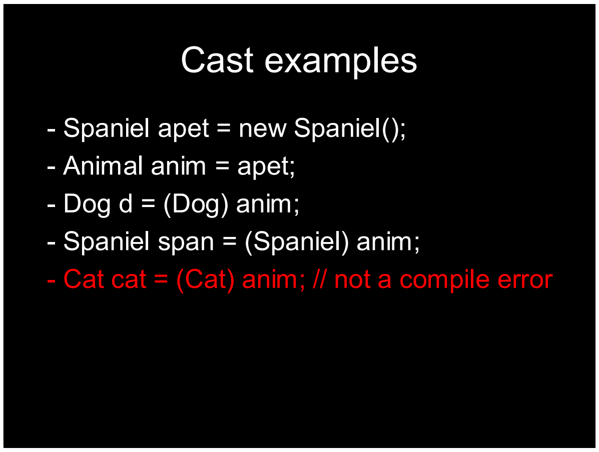
Duck

Spaniel



Cast examples

- Can always cast upward through direct and indirect hierarchy.
  - – Spaniel apet = new Spaniel();
  - – Animal anim = apet;
- You can cast **down** a hierarchy also, but you must always explicitly write the cast and the object must be a legitimate instance of the class you are casting to.



Cast examples

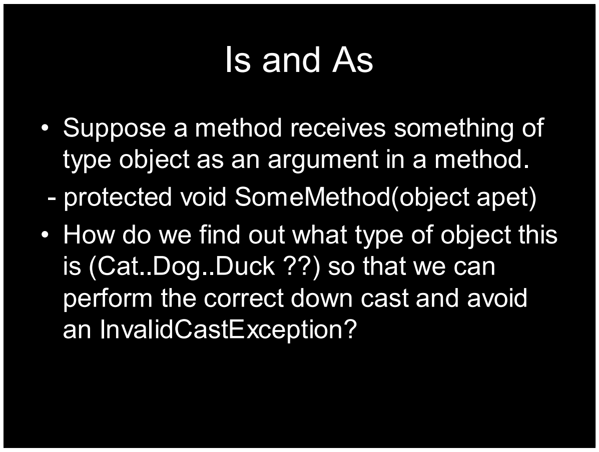
- Spaniel apet = new Spaniel();

- Animal anim = apet;

- Dog d = (Dog) anim;

- Spaniel span = (Spaniel) anim;

- Cat cat = (Cat) anim; // not a compile error

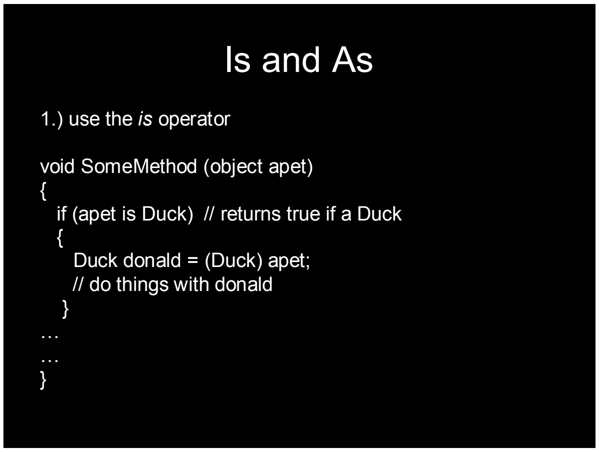


Is and As

- Suppose a method receives something of type object as an argument in a method.

 - protected void SomeMethod(object apet)

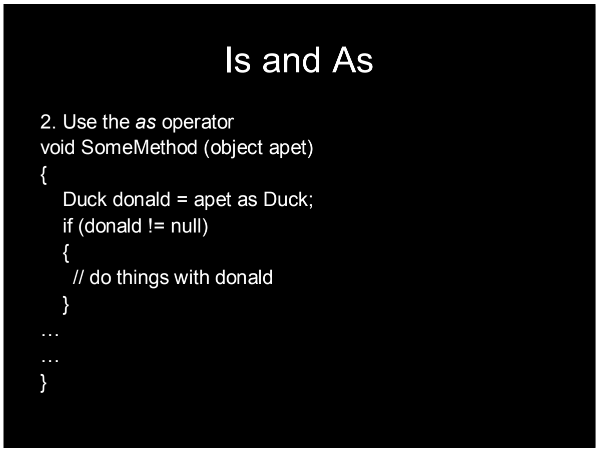
- How do we find out what type of object this is (Cat..Dog..Duck ??) so that we can perform the correct down cast and avoid an InvalidCastException?



Is and As

1.) use the *is* operator

```
void SomeMethod (object apet)
{
   if (apet is Duck)  // returns true if a Duck
   {
      Duck donald = (Duck) apet;
      // do things with donald
    }
…
…
}
```



Is and As

2. Use the *as* operator

```
void SomeMethod (object apet)
{
    Duck donald = apet as Duck;
    if (donald != null)
    {
      // do things with donald
    }
…
…
}
```