Anglia Ruskin University

Private attributes and Public methods

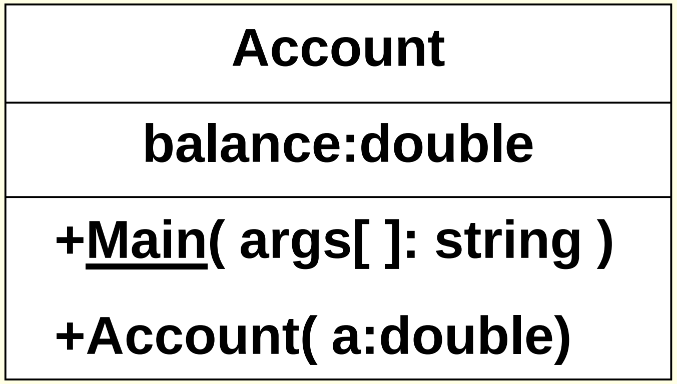
scope

- determines the system visibility of variables

{…}	*delimits scope of variables*
class A {...}	*class scope*
public void method() {...}	*method scope*
if (condition) {...}	*local (control structure) scope*

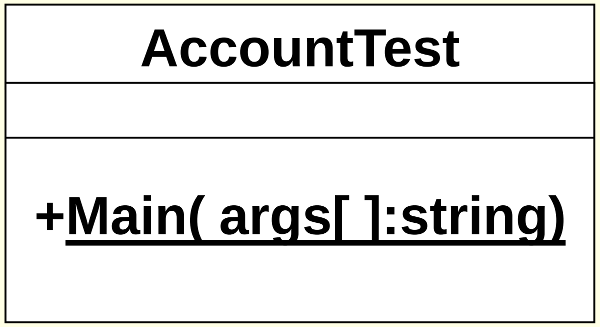
Also a scope terminology to emphasise attribute/method accessibility within/between classes

public	**+**	*Directly accessible outside class (public)*
protected	**#**	*Visible throughout inheritance tree*
no designation		*Effectively private in C#*
private	**-**	*Indirectly accessible outside class*

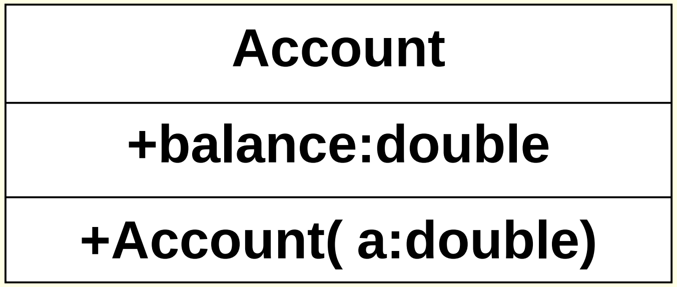


```
// File: Account.cs
class Account
{
   double balance;

   public static void Main( string[ ] args)
   {
      Account acc1 = new Account( 400.0 );
      Console.WriteLine(“Balance is “+ acc1.balance);
   }

   public Account( double a)
   {
      balance = a;
   }
}
```

```
class AccountTest
{
   public static void Main( string[ ] args)
   {
       Account acc1 = new Account( 400.0 );
       Console.WriteLine(“Balance is “+ acc1.balance);
   }
}
```



```
class Account
{
  public double balance;

  public Account( double a)
  {
      balance = a;
  }
}
```



Attributes *if public* can be accessed directly

This is not normally good practice

So reduce attribute scope by making the data private

```
class Account
{
  private double balance;

  public Account( double input )
  {
    balance = a;
  }
}
```

Account
- balance:double
+Account(a:double)

In Main() :

```
Console.WriteLine(“Balance is “+ acc2.balance );
```

This will not now work!

Visible in class *and* *outside* using class or object name

```
class Classname
{
   private an_attribute;

   public a_Method()
   {
     // ..
   }
}
```

Visible *only* in class or object

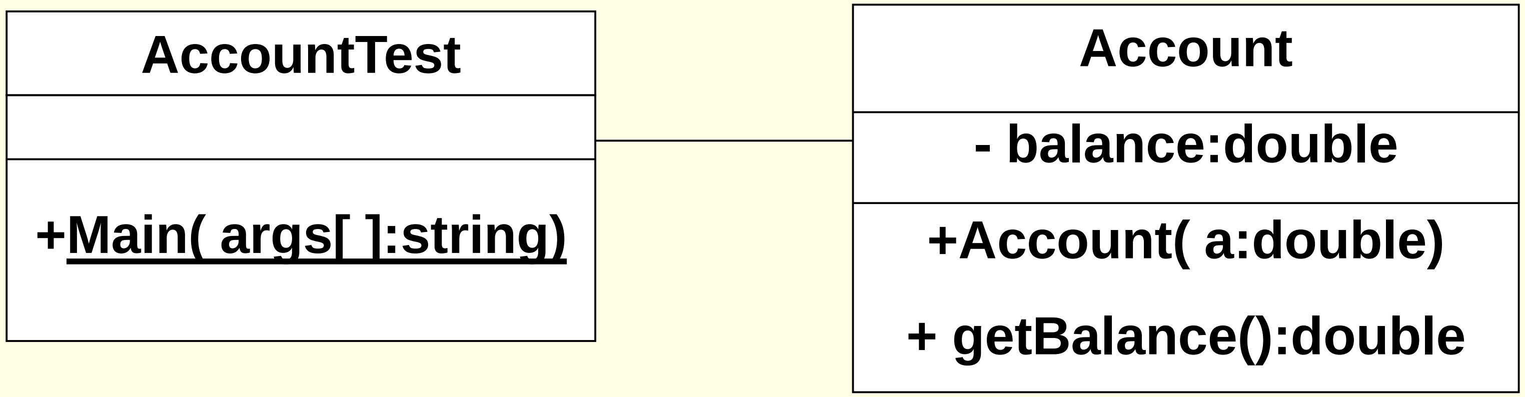
Still need a way of accessing private data from outside a class - use a *public method*

```
public double getBalance( )
{
      return balance;
}
```

This is a ‘**get**’ method or ‘**accessor**’ method

```
class Account
{
  private double balance;

  public Account( double input )
  {
    balance = input;
  }

  public double getBalance()
  {
    return balance;
  }
}
```

Usually have a public ‘get’ method for each private data



In Main() :

```
Console.WriteLine("Balance is "+
                         acc2.getBalance() );
```

Anglia Ruskin University

To *change* private data from outside a class – again use a public method

```
public void setBalance( double bal )
{
      balance  =  bal;
}
```

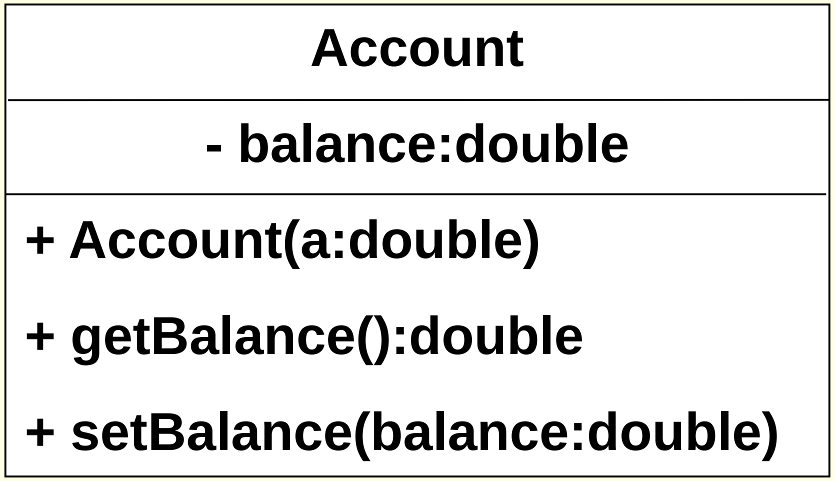
Called a ‘**set**’ method or ‘**mutator**’ method

In Main() :

```
            acc1.setBalance( 125.0);
```

Anglia Ruskin University

```
class Account
{
   private double balance;

   public Account( double a )
   {
      balance = a;
   }

   public double getBalance()
   {
      return balance;
   }

   public void setBalance(double balance)
   {
       this.balance = balance;
   }
}
```



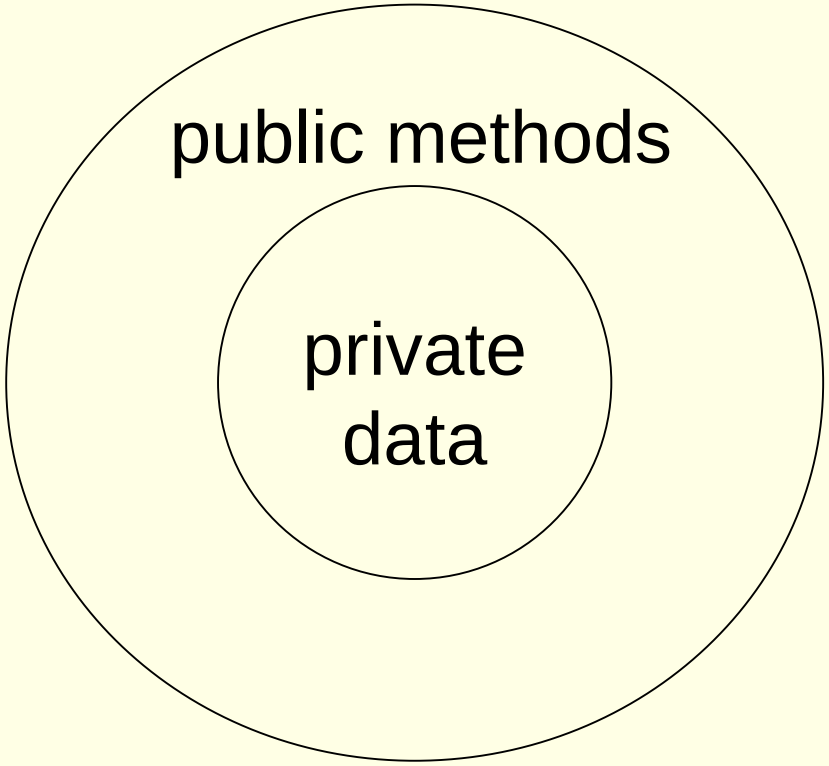
Anglia Ruskin University

```
class AccountTest
{
   public static void Main( string[ ] args)
   {
       Account acc1 = new Account(550.0);
       Console.WriteLine( “Balance of acc1 is “ +
                                         acc1.getBalance() );
       Account acc2 = new Account(0.0);
       acc2.setBalance( 330.0);
       Console.WriteLine( “Balance of acc2 is “+
                                        acc2.getBalance() );
   }
}
```

```
Balance of acc1 is 550.0
Balance of acc2 is 330.0
```

Anglia Ruskin University

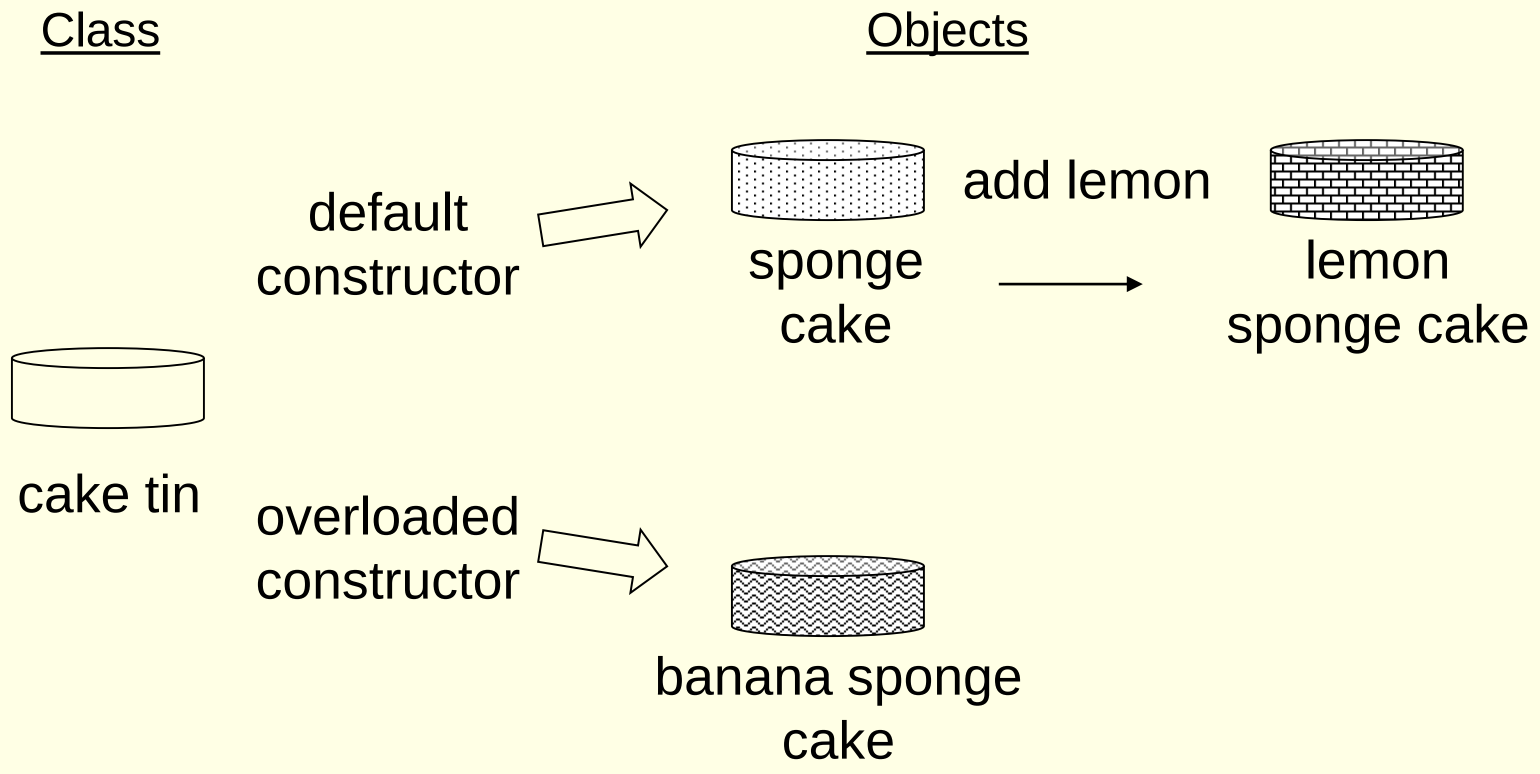
Data Hiding



Private data can only be accessed via an interface of public methods

Anglia Ruskin University

A class is like a ‘template’ but can still do things


Class
Objects
default constructor
sponge cake
add lemon
lemon sponge cake
cake tin
overloaded constructor
banana sponge cake