Overriding or Overloading Methods

Overriding a method

A method is **overridden** within an inheritance hierarchy when there are two methods with the same name, same return type and the same number of input parameters.

The method in the base class is prefixed with the modifier keyword **virtual** (if it will be implemented) or by **abstract** (if it will never be implemented).
The method in the subclass is prefixed by the **override** modifier.

The compiler decides which method to use by looking at the object that has called the method.

```
//base class Player
       public abstract void Draw();      //an abstract method has no body

or
       public virtual void Draw()
       {
       …….
       }


//child class Sprite

       public override void Draw()
        {
            left = StartPosition.X;
            top = StartPosition.Y;
            width = Image.Width;
            height = Image.Height;
        }
```

Then:

`player.Draw()` would use the base class version of the Draw()method (if it is virtual!)
`sprite.Draw()` would use the subclass version of the Draw() method.

Overloading a method

A method is **overloaded** when there are two methods with the same name, same return type but different input parameters.

The compiler decides which method to use by counting the number of parameters and checking the data types that are provided when the method is called.

For example, `random.Next(7)` and `random.Next (1,7)` show the Next method has been overloaded.

For example, Constructors can be overloaded:
You can have two constructors called Sprite with differing numbers of input parameters.

```
        public Sprite(Point location, Size size)
        {
            this.location = location;
            this.size = size;
        }
```

and

```
        public Sprite(string spriteFile, Point location, Size size)
              :base(location, size)
        {
            image = Image.FromFile(spriteFile);
            boundary = new Rectangle(location, size);
        }
```

Either version of the Sprite constructor can be used. The compiler uses the version with the matching parameter list.