App-04 - Inheritance

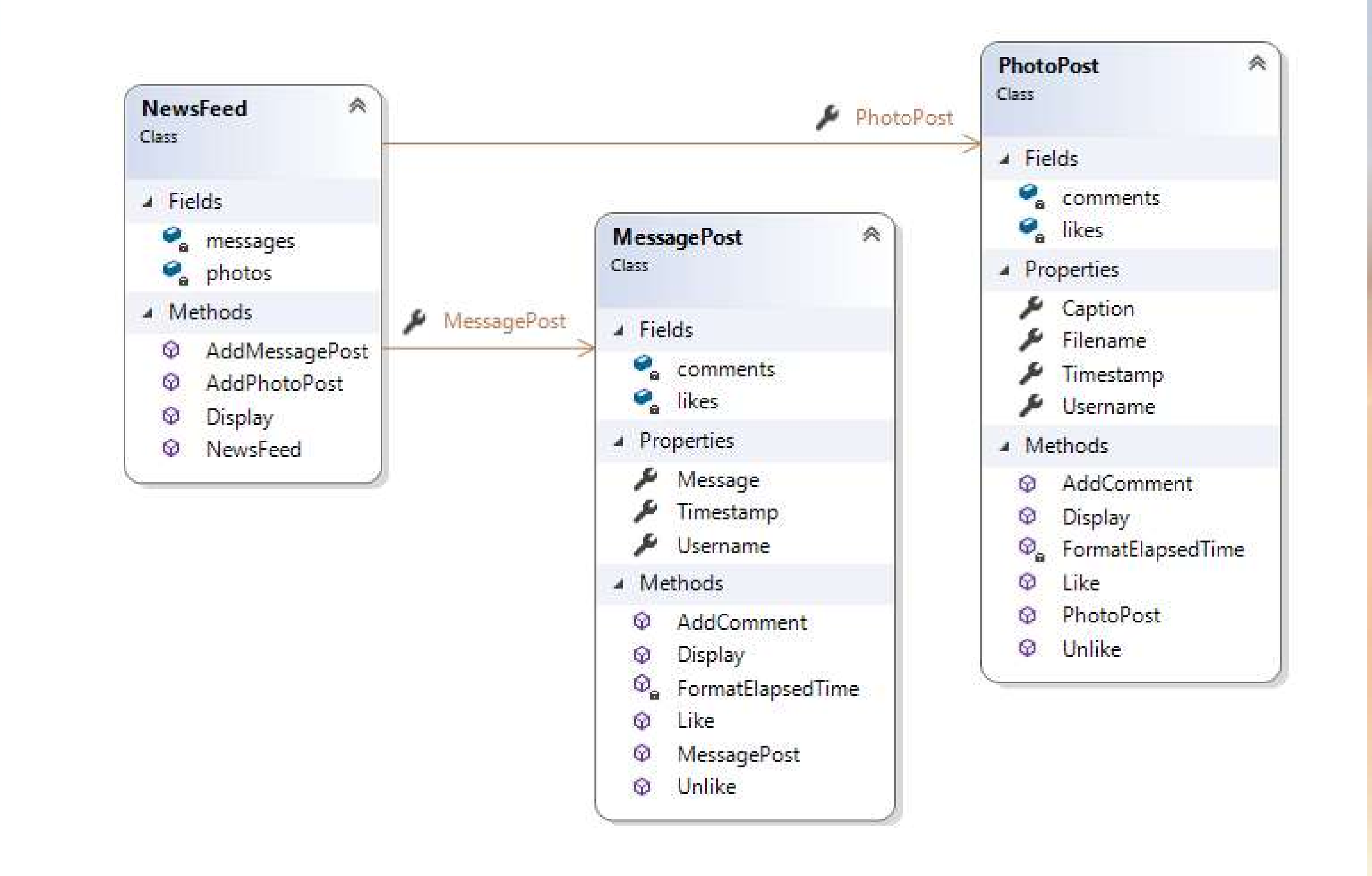
CO453 Applications Programming

Network (see BlueJ Chapter 8 v5)

1. Add message posts
2. Add photo posts
3. The posts should be saved (in memory)
4. Display all posts for a time period
5. Display posts for an Author
6. Remove a post
7. Add comments to posts
8. Like and unlike posts

Similar to FaceBook

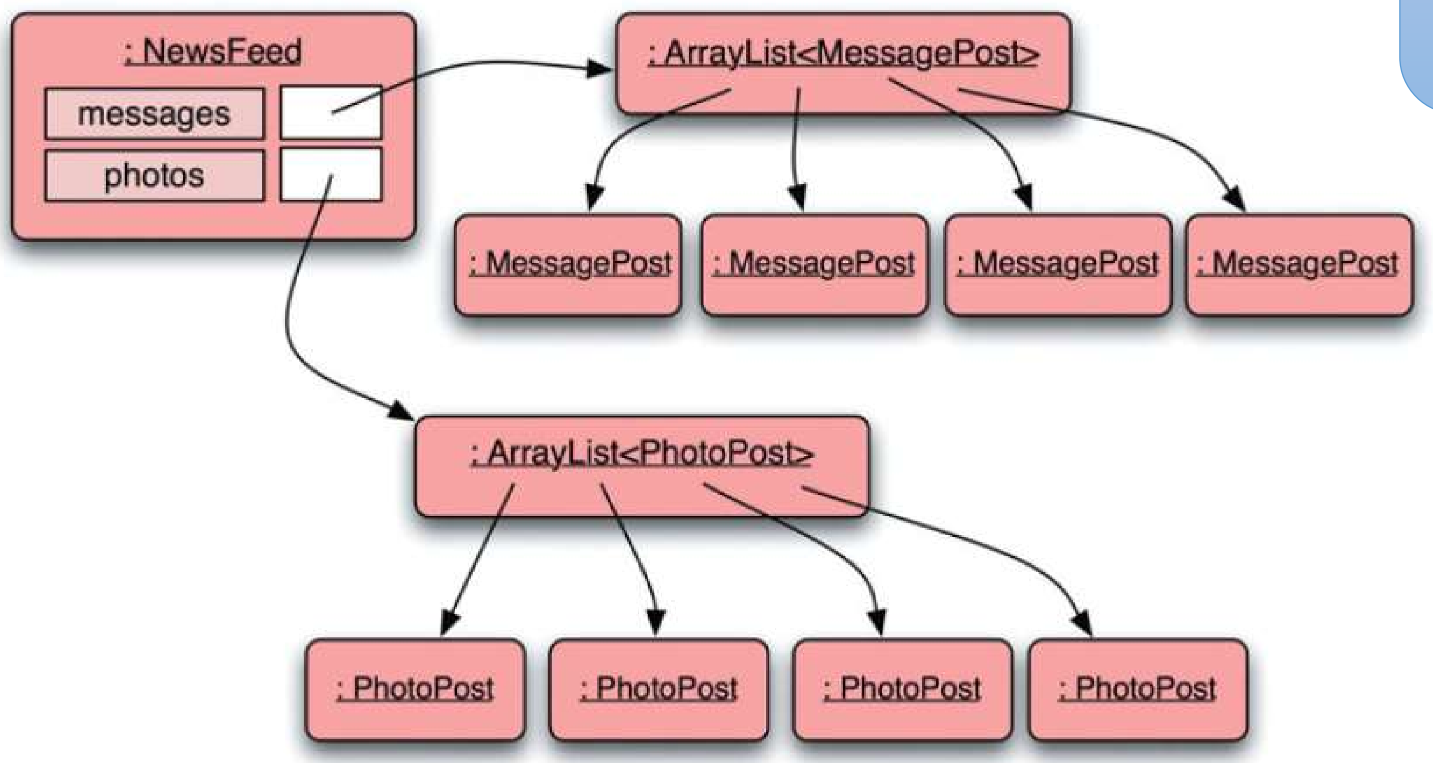
Initial Design (App04 v0.1)



Notice the Code Duplication!

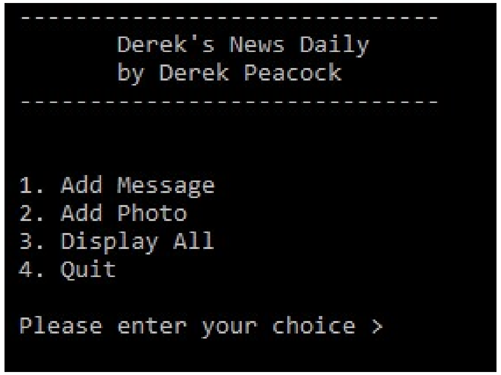
What do MessagePost and PhotoPost have in common?

Objects

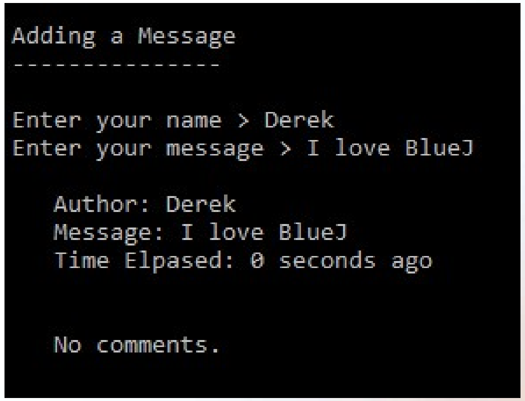


What is the issue with having separate array lists for messages and photos?

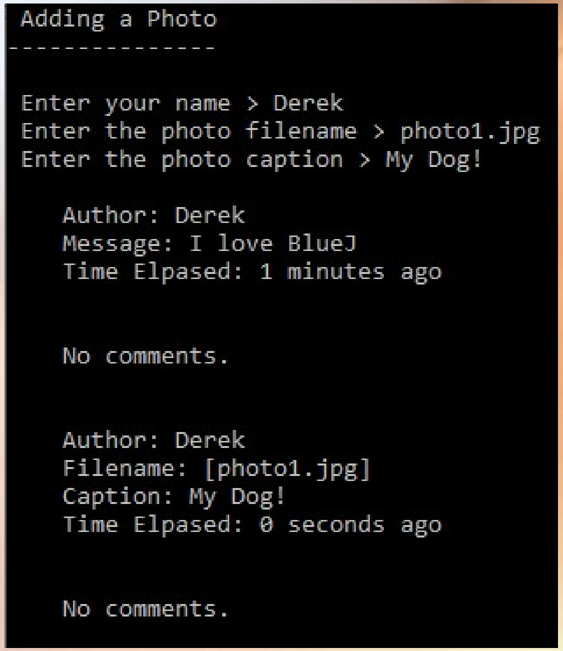
Current Features (version 0.1)



```
------------------------------
       Derek's News Daily
       by Derek Peacock
------------------------------


1. Add Message
2. Add Photo
3. Display All
4. Quit

Please enter your choice >
```



```
Adding a Message
---------------

Enter your name > Derek
Enter your message > I love BlueJ

   Author: Derek
   Message: I love BlueJ
   Time Elpased: 0 seconds ago


   No comments.
```



```
 Adding a Photo
---------------

 Enter your name > Derek
 Enter the photo filename > photo1.jpg
 Enter the photo caption > My Dog!

    Author: Derek
    Message: I love BlueJ
    Time Elpased: 1 minutes ago


    No comments.


    Author: Derek
    Filename: [photo1.jpg]
    Caption: My Dog!
    Time Elpased: 0 seconds ago


    No comments.
```

What if we want to add another kind of post, e.g ActivityPost?

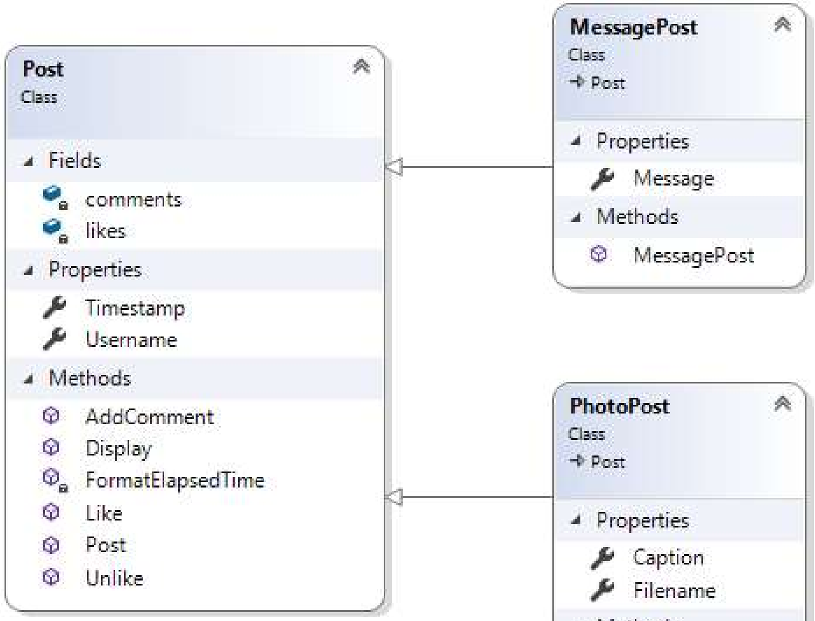
Inheritance – A general Post class

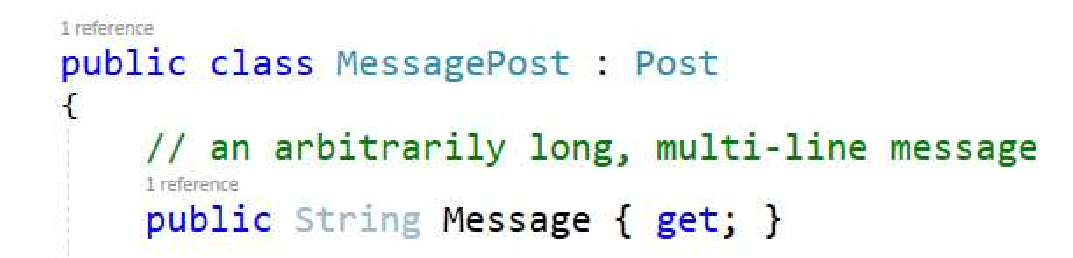
Super/base class or parent

Sub class or child

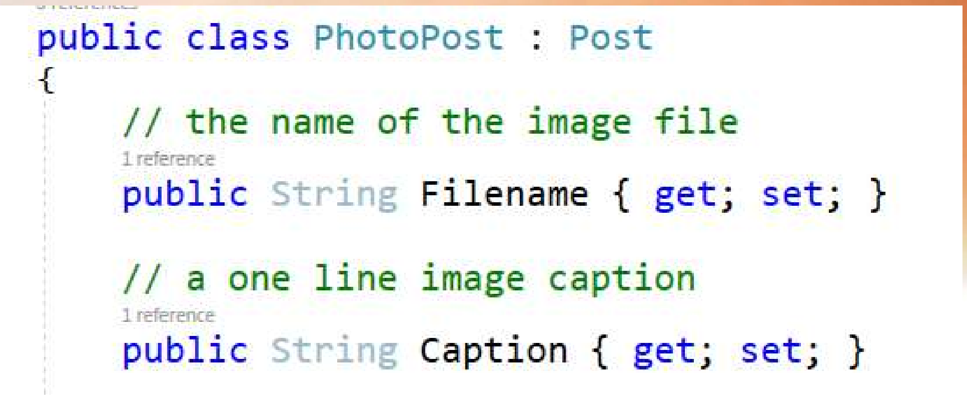
MessagePost **is a kind of** Post

PhotoPost **is a kind of** Post





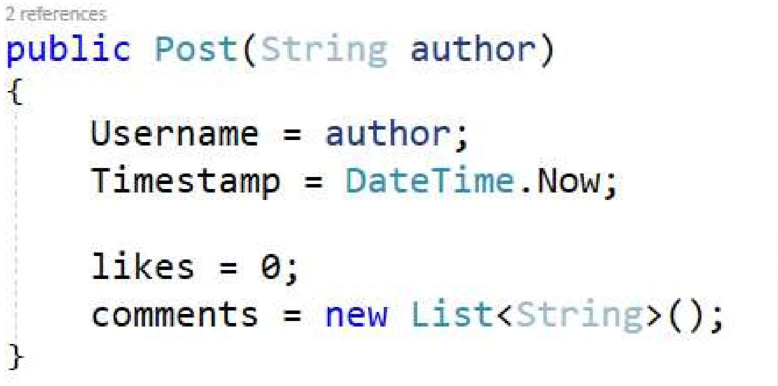
```
1 reference
public class MessagePost : Post
{
    // an arbitrarily long, multi-line message
    1 reference
    public String Message { get; }
```



```
public class PhotoPost : Post
{
    // the name of the image file
    1 reference
    public String Filename { get; set; }

    // a one line image caption
    1 reference
    public String Caption { get; set; }
```

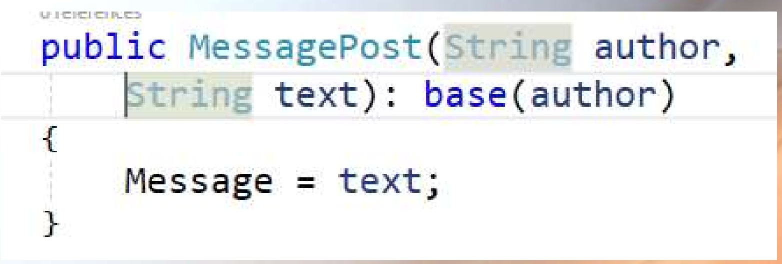
Class Constructors

Single parameter



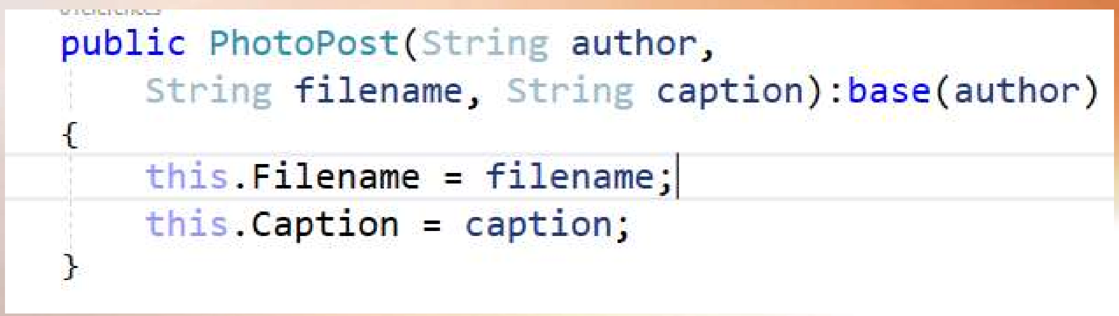
```
2 references
public Post(String author)
{
    Username = author;
    Timestamp = DateTime.Now;

    likes = 0;
    comments = new List<String>();
}
```

2 parameters with **author** passed to the **base** class



```
public MessagePost(String author,
    String text): base(author)
{
    Message = text;
}
```

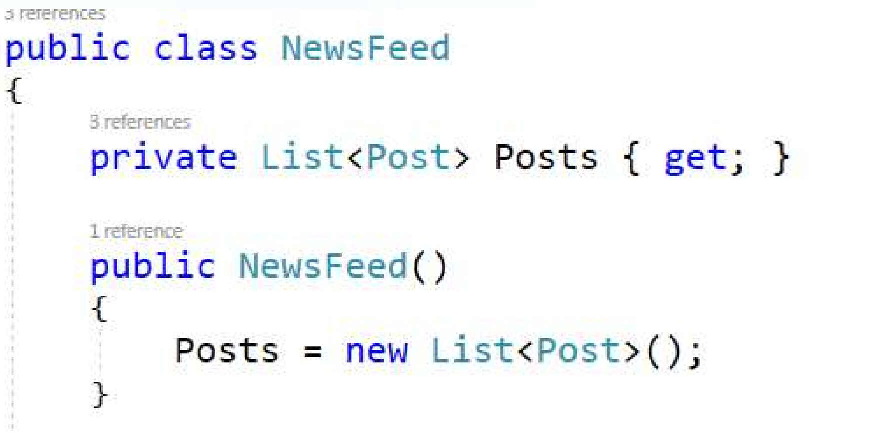
3 parameters with **author** passed to the **base** class



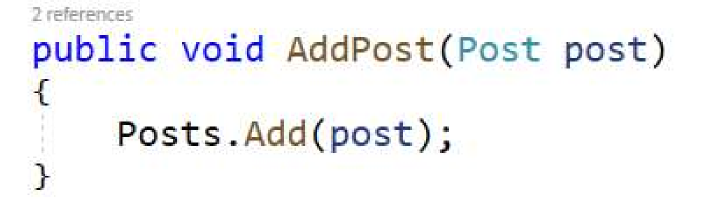
```
public PhotoPost(String author,
    String filename, String caption):base(author)
{
    this.Filename = filename;
    this.Caption = caption;
}
```

NewsFeed using Post

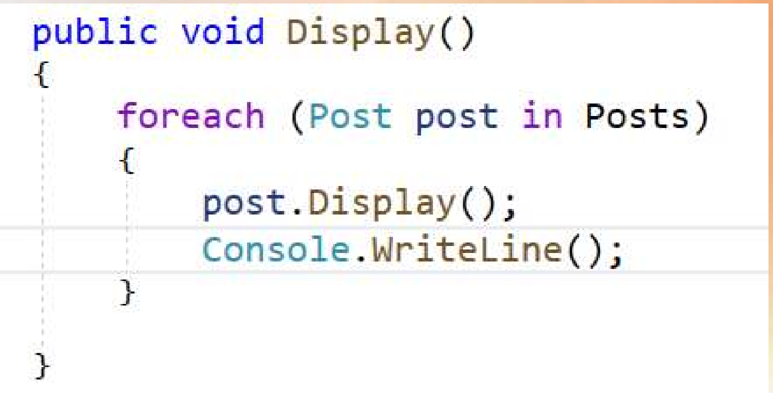
All posts are now listed in time order



```
3 references
public class NewsFeed
{
    3 references
    private List<Post> Posts { get; }

    1 reference
    public NewsFeed()
    {
        Posts = new List<Post>();
    }
```



```
2 references
public void AddPost(Post post)
{
    Posts.Add(post);
}
```

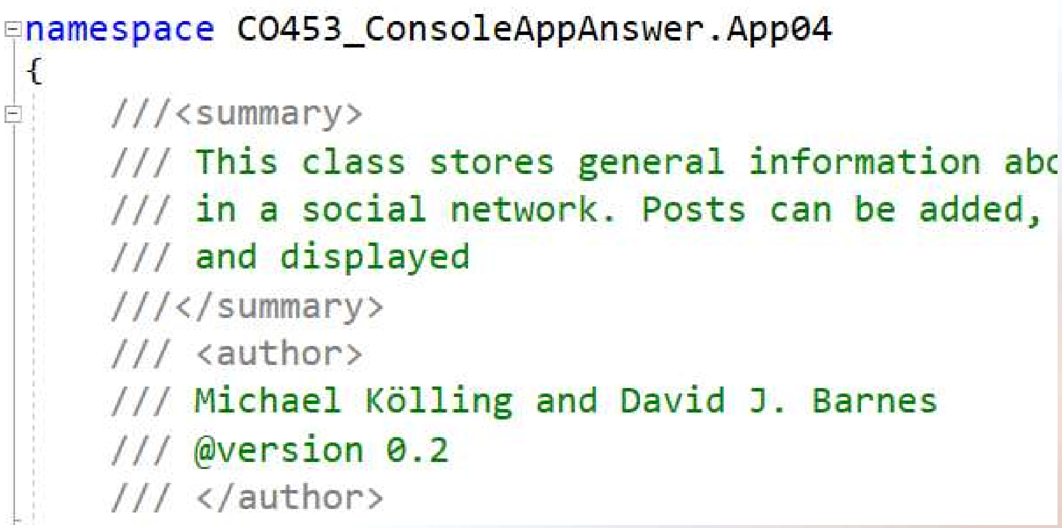


```
public void Display()
{
    foreach (Post post in Posts)
    {
        post.Display();
        Console.WriteLine();
    }

}
```

NewsFeed is much simplified with only a single List of **Posts**

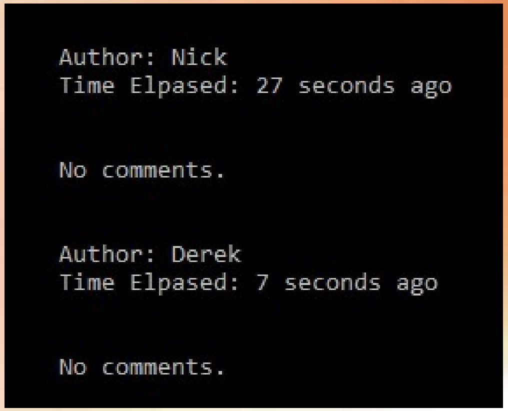
Adding a User Interface

Namespace can be used to create separate areas



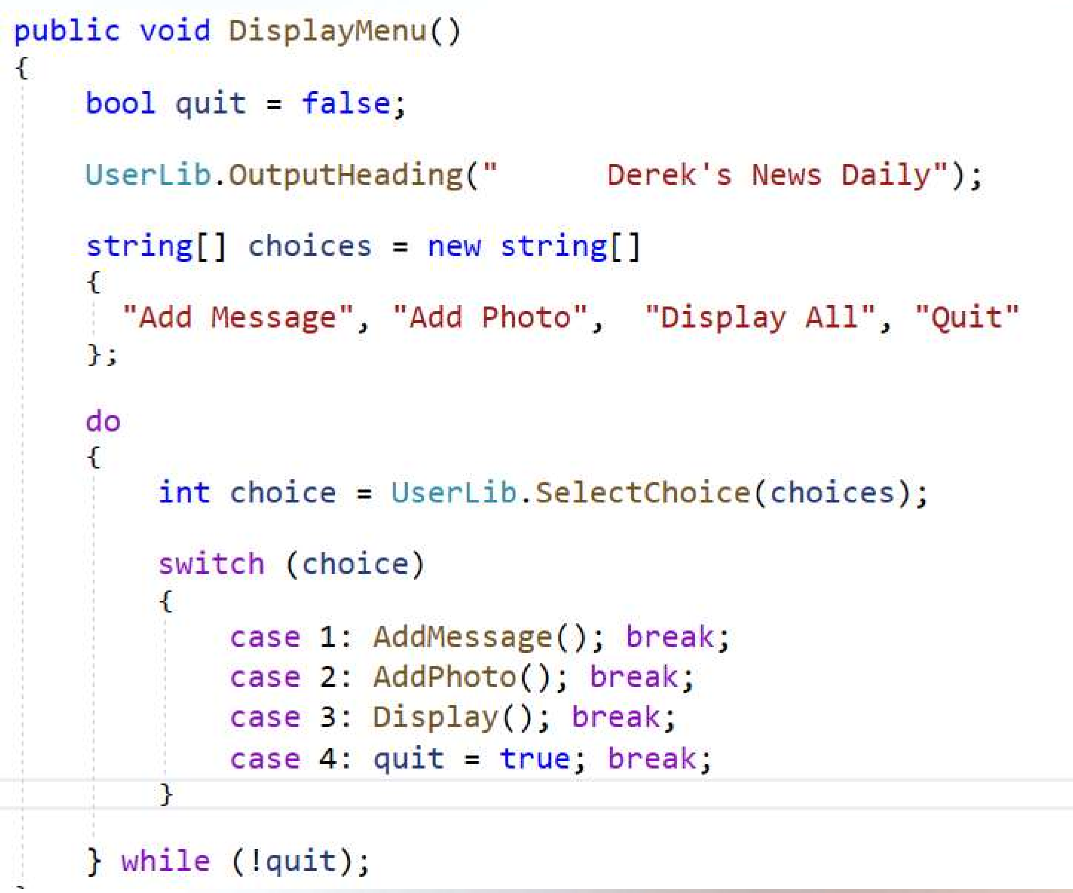
```
namespace CO453_ConsoleAppAnswer.App04
{
    ///<summary>
    /// This class stores general information abc
    /// in a social network. Posts can be added,
    /// and displayed
    ///</summary>
    /// <author>
    /// Michael Kölling and David J. Barnes
    /// @version 0.2
    /// </author>
```

The Display method in Post does not know about messages or photos

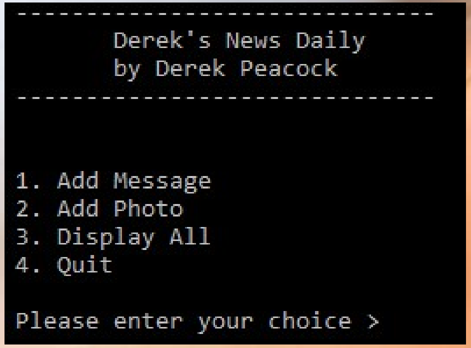


```
Author: Nick
Time Elpased: 27 seconds ago


No comments.


Author: Derek
Time Elpased: 7 seconds ago


No comments.
```

NetworkApp DisplayMenu

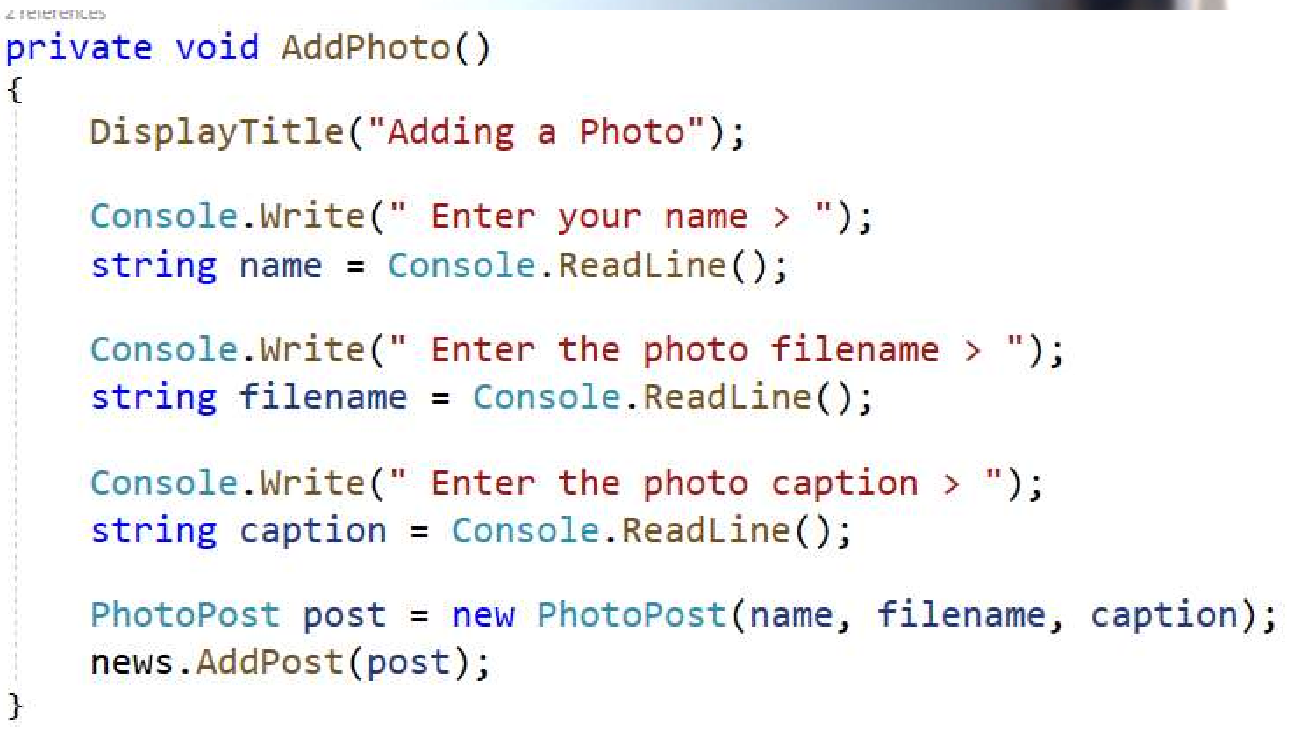


```
public void DisplayMenu()
{
    bool quit = false;

    UserLib.OutputHeading("      Derek's News Daily");

    string[] choices = new string[]
    {
     "Add Message", "Add Photo",  "Display All", "Quit"
    };

    do
    {
        int choice = UserLib.SelectChoice(choices);

        switch (choice)
        {
            case 1: AddMessage(); break;
            case 2: AddPhoto(); break;
            case 3: Display(); break;
            case 4: quit = true; break;
        }

    } while (!quit);
```

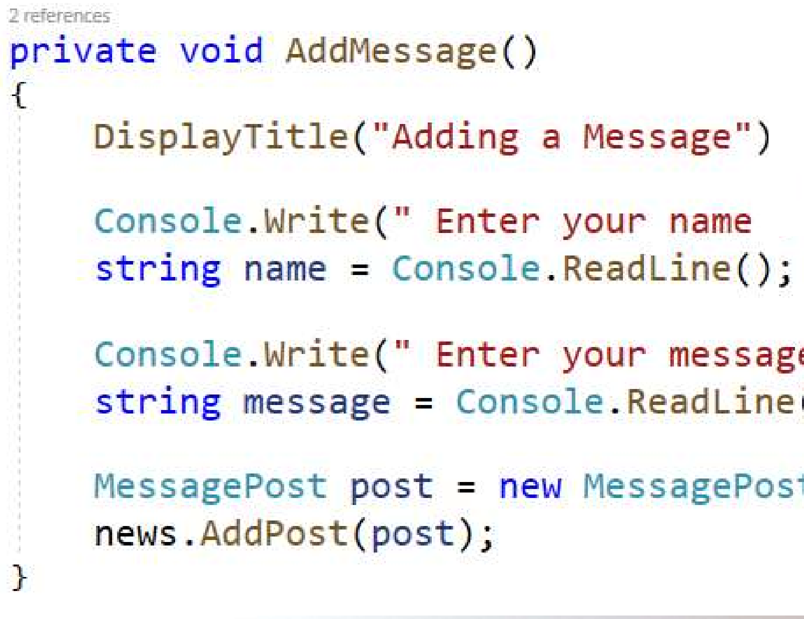


```
------------------------------
       Derek's News Daily
       by Derek Peacock
------------------------------


1. Add Message
2. Add Photo
3. Display All
4. Quit

Please enter your choice >
```

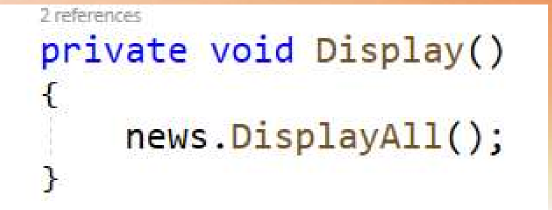
Adding Posts



```
2 references
private void AddPhoto()
{
    DisplayTitle("Adding a Photo");

    Console.Write(" Enter your name > ");
    string name = Console.ReadLine();

    Console.Write(" Enter the photo filename > ");
    string filename = Console.ReadLine();

    Console.Write(" Enter the photo caption > ");
    string caption = Console.ReadLine();

    PhotoPost post = new PhotoPost(name, filename, caption);
    news.AddPost(post);
}
```



```
2 references
private void AddMessage()
{
    DisplayTitle("Adding a Message")

    Console.Write(" Enter your name
    string name = Console.ReadLine();

    Console.Write(" Enter your message > ");
    string message = Console.ReadLine();

    MessagePost post = new MessagePost(name, message);
    news.AddPost(post);
}
```

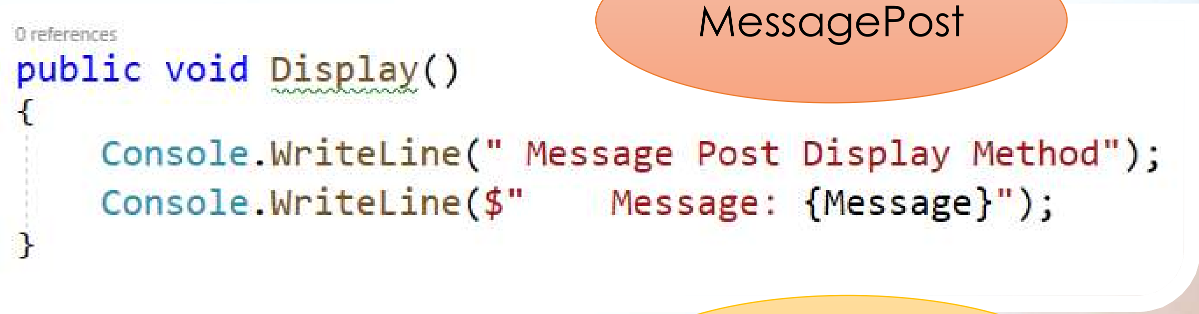


```
2 references
private void Display()
{
    news.DisplayAll();
}
```

Polymorphism

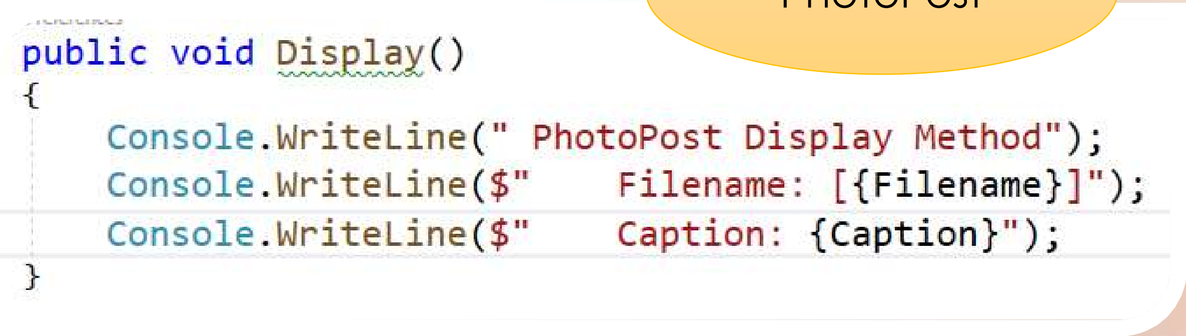
Only Post.Display() used

MessagePost

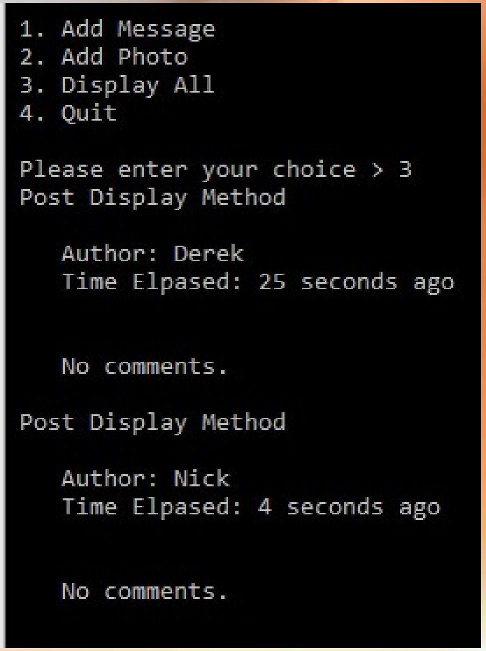


```
0 references
public void Display()
{
    Console.WriteLine(" Message Post Display Method");
    Console.WriteLine($"    Message: {Message}");
}
```

PhotoPost

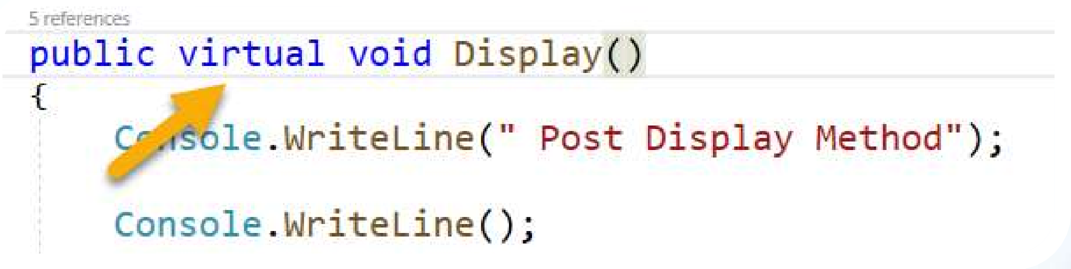


```
public void Display()
{
    Console.WriteLine(" PhotoPost Display Method");
    Console.WriteLine($"    Filename: [{Filename}]");
    Console.WriteLine($"    Caption: {Caption}");
}
```

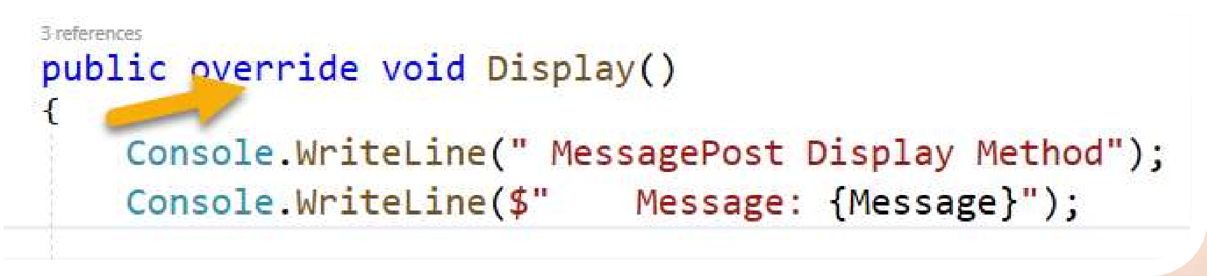


```
1. Add Message
2. Add Photo
3. Display All
4. Quit

Please enter your choice > 3
Post Display Method

   Author: Derek
   Time Elpased: 25 seconds ago


   No comments.

Post Display Method

   Author: Nick
   Time Elpased: 4 seconds ago


   No comments.
```

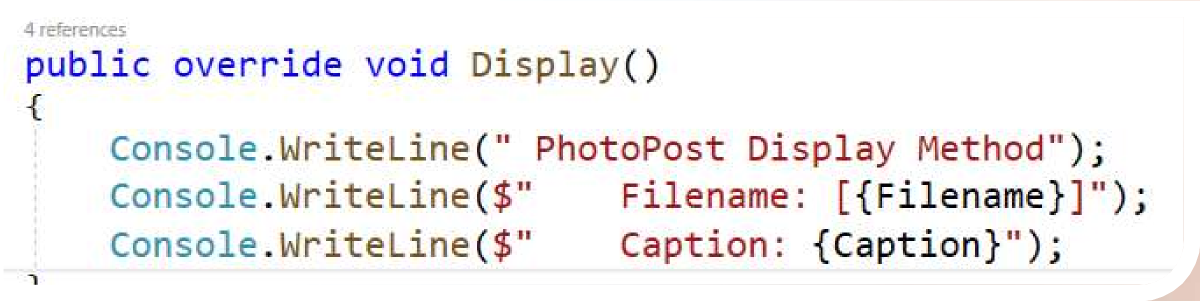
Overriding Methods



```
5 references
public virtual void Display()
{
    Console.WriteLine(" Post Display Method");

    Console.WriteLine();
```

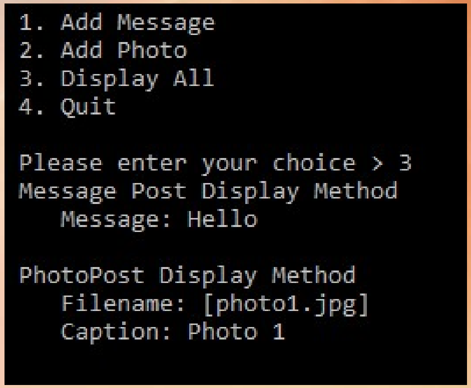


```
3 references
public override void Display()
{
    Console.WriteLine(" MessagePost Display Method");
    Console.WriteLine($"    Message: {Message}");
```



```
4 references
public override void Display()
{
    Console.WriteLine(" PhotoPost Display Method");
    Console.WriteLine($"    Filename: [{Filename}]");
    Console.WriteLine($"    Caption: {Caption}");
```

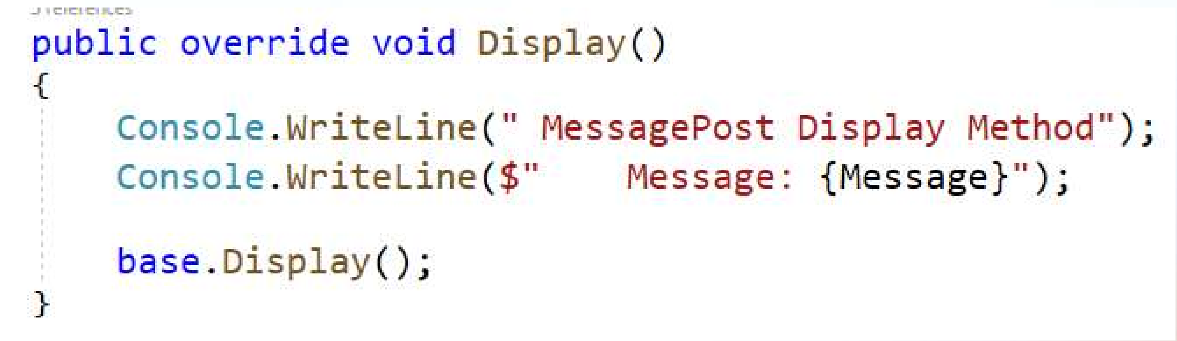
Sub class Display() is used but The common **Post** properties are not displayed



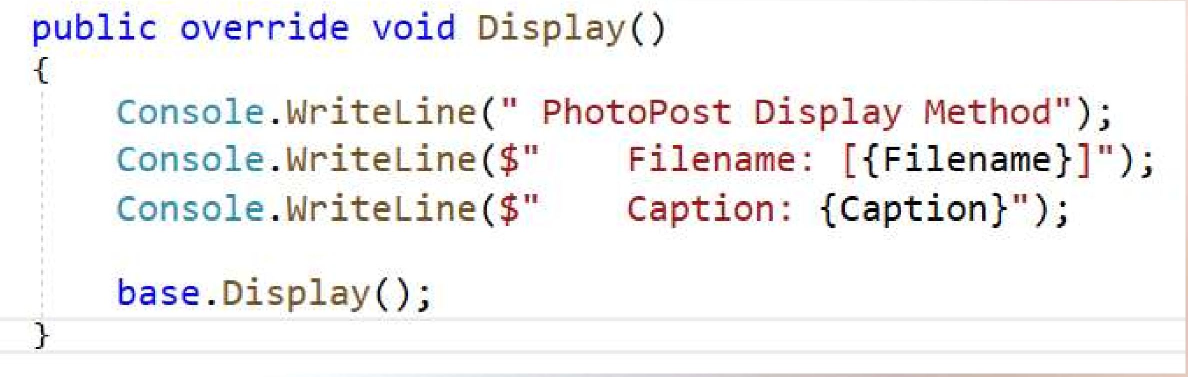
```
1. Add Message
2. Add Photo
3. Display All
4. Quit

Please enter your choice > 3
Message Post Display Method
   Message: Hello

PhotoPost Display Method
   Filename: [photo1.jpg]
   Caption: Photo 1
```

Using base Display

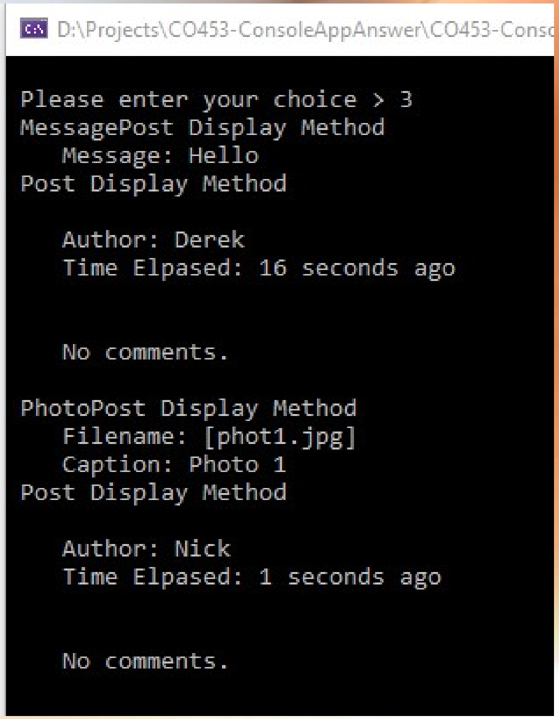
All three Display methods are now used



```
public override void Display()
{
    Console.WriteLine(" MessagePost Display Method");
    Console.WriteLine($"    Message: {Message}");

    base.Display();
}
```



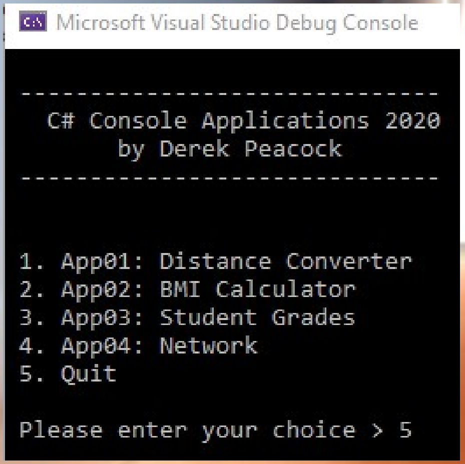
```
public override void Display()
{
    Console.WriteLine(" PhotoPost Display Method");
    Console.WriteLine($"    Filename: [{Filename}]");
    Console.WriteLine($"    Caption: {Caption}");

    base.Display();
}
```



```
Please enter your choice > 3
MessagePost Display Method
   Message: Hello
Post Display Method

   Author: Derek
   Time Elpased: 16 seconds ago


   No comments.

PhotoPost Display Method
   Filename: [phot1.jpg]
   Caption: Photo 1
Post Display Method

   Author: Nick
   Time Elpased: 1 seconds ago


   No comments.
```

Week 4 Independent Study

- Complete as many features as possible
  1. Add Message posts
  2. Add Photo posts
  3. The posts should be saved (memory only)
  4. Display all posts in time order
  5. Display posts for a username
  6. Remove posts
  7. Add comments to posts
  8. Like and unlike posts



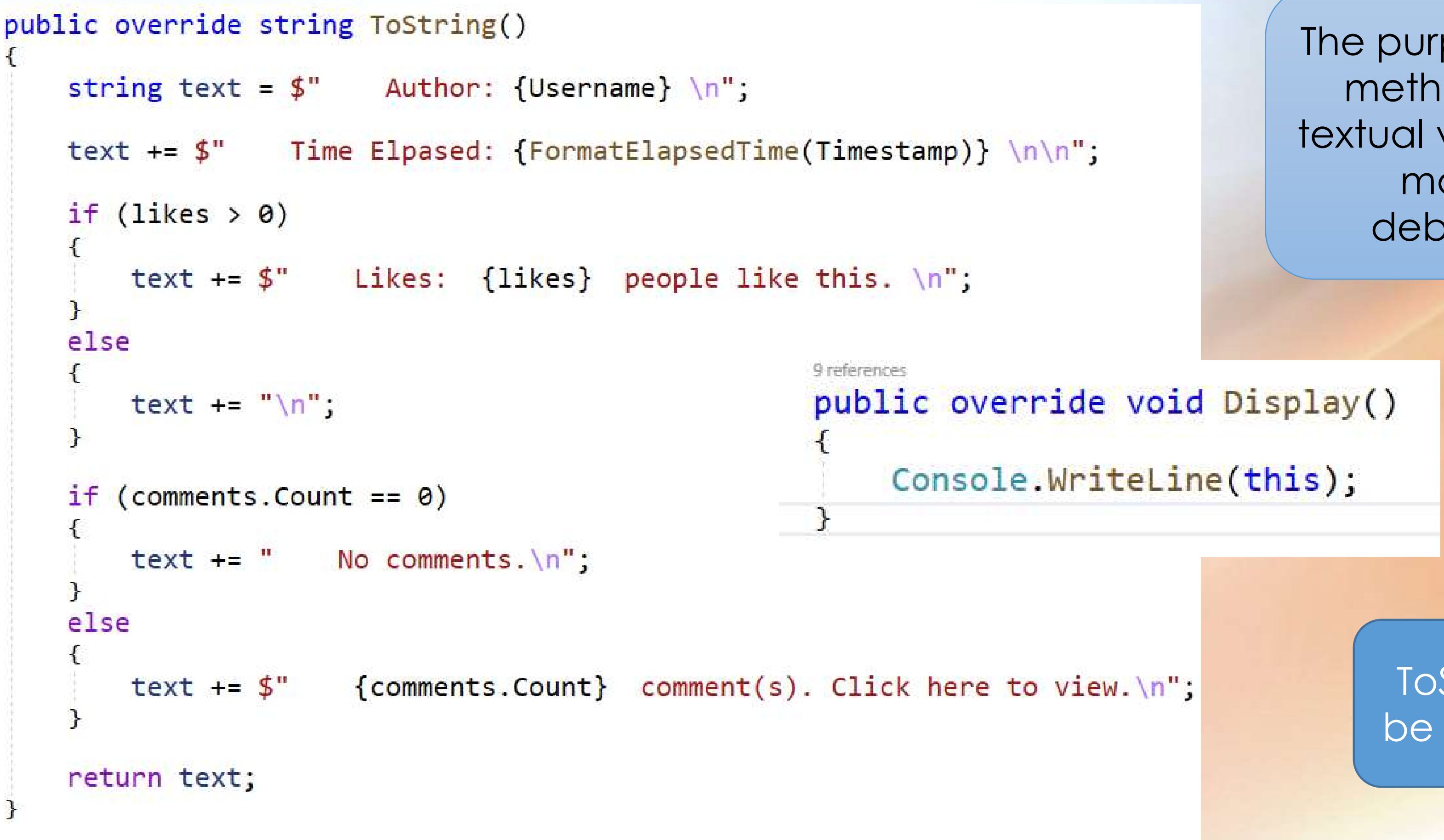
```
------------------------------
  C# Console Applications 2020
       by Derek Peacock
------------------------------


1. App01: Distance Converter
2. App02: BMI Calculator
3. App03: Student Grades
4. App04: Network
5. Quit

Please enter your choice > 5
```

These features are much more difficult to implement without a GUI (graphics user interface)

Do your best!

Overriding ToString()

All objects inherit a ToString()



```
public override string ToString()
{
    string text = $"    Author: {Username} \n";

    text += $"    Time Elapsed: {FormatElapsedTime(Timestamp)} \n\n";

    if (likes > 0)
    {
        text += $"    Likes:  {likes}  people like this. \n";
    }
    else
    {
        text += "\n";
    }

    if (comments.Count == 0)
    {
        text += "    No comments.\n";
    }
    else
    {
        text += $"    {comments.Count}  comment(s). Click here to view.\n";
    }

    return text;
}
```

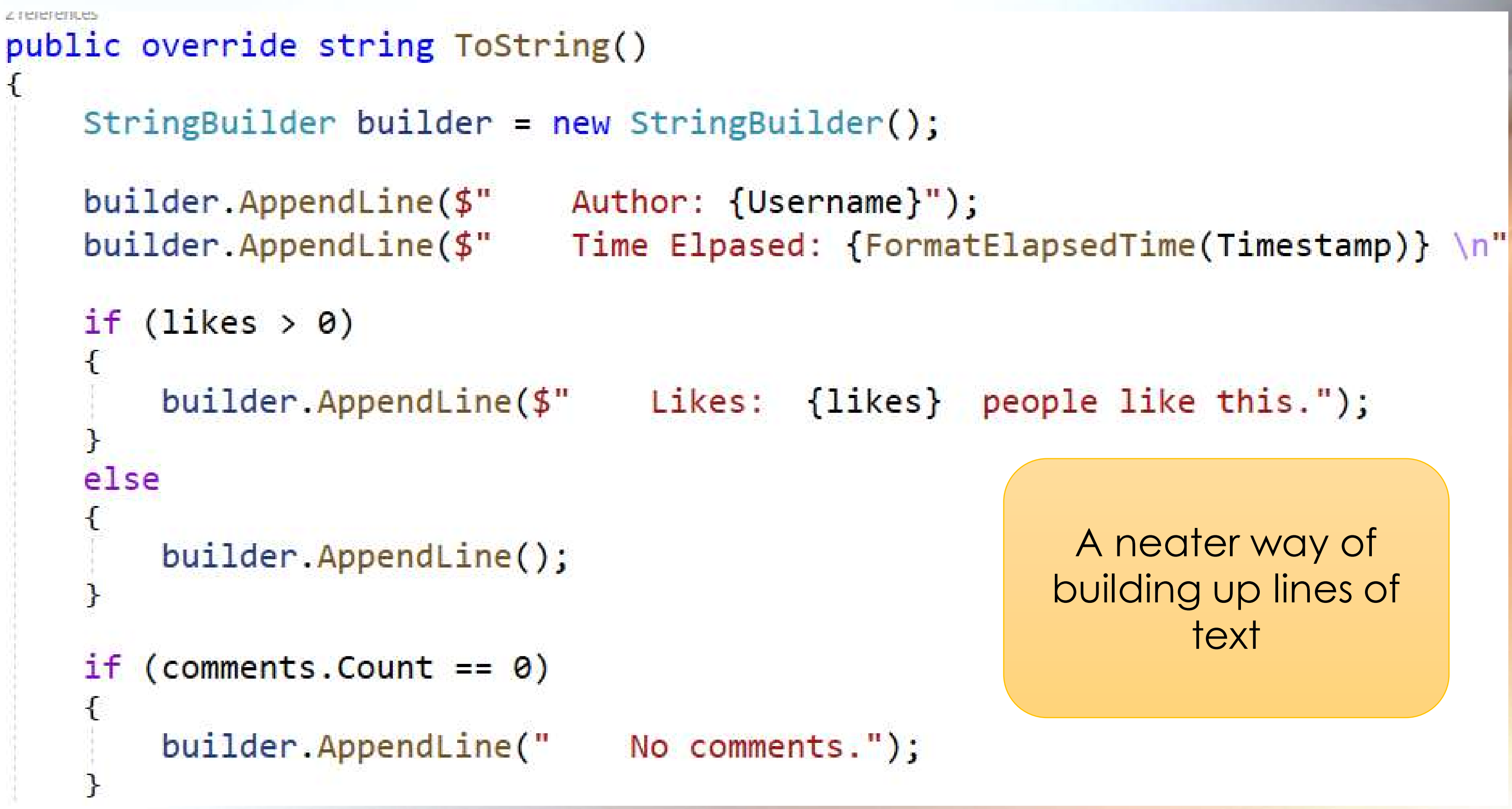
```
9 references
public override void Display()
{
    Console.WriteLine(this);
}
```

The purpose of the **ToString**() method is to produce a textual version of the objects main variables for debugging purposes

ToString() can then be used by Display()

StringBuilder Class

ToString() can be added to MessagePost and PhotoPost



2 references

```
public override string ToString()
{
    StringBuilder builder = new StringBuilder();

    builder.AppendLine($"    Author: {Username}");
    builder.AppendLine($"    Time Elpased: {FormatElapsedTime(Timestamp)} \n"

    if (likes > 0)
    {
        builder.AppendLine($"    Likes:  {likes}  people like this.");
    }
    else
    {
        builder.AppendLine();
    }

    if (comments.Count == 0)
    {
        builder.AppendLine("    No comments.");
    }
```

A neater way of building up lines of text