COMMENTS, STYLE and SYNTAX

- Each computer programming language has rules which govern how a program can be written. These rules are called the format or syntax of the language. If you do not follow the syntax rules of the programming language, your program will not compile or run.

COMMENTS

- Comments are used to explain and clarify program code for other programmers and have no effect on the way an application runs. Comments should be used whenever code may be ambiguous, but not reiterate what is clear form the code.
- The C# language uses the "//" characters to identify a comment line.
- You can also add a comment at the end of a line of code:

```
// This is a comment by itself on a line.
int height = 32;    //this is a comment at the end of a source line
```

MULTI LINE COMMENTS

- To create comments across multiple lines, you can surround the entire multi-line comment with the "/* to start the comment block and "*/" to end the comment block.

COMMENT HEADER

- Each form's code must have a comment header which includes the name of the programmer, the name of the program, the date and a brief description of what the program (or form) does.

```
/* Created by: S. Myers
 * Program name: Assignment 1 – Planets
 * Date created: September 27
 * Description: Displays a picture and info on the planets in our solar system
 */
```

THE .NET FRAMEWORK

- The .NET Framework is a giant library of pre-built methods and components. The .NET Framework objects in C# are carefully organized into groups called "namespaces". The System namespace is the most common namespace in the C# language. When the IDE creates a project for you, it will automatically add several **using** statements at the top of each source file to give you access to the necessary form and other objects.

STATEMENTS AND WHITE-SPACES

- A statement is a piece of code that performs some specific action.

```
height = 32;
```

- You could type multiple statements (or even your entire program!) together on the same line of text. The compiler doesn't need the code to look pretty; it just insists that you obey the rules for correctly formatting statements.

```
height = 32; length = 21;
```

- We use white-space to make the code easier to read. The compiler doesn't care, it ignores white-spaces including returns. The C# language uses a semi-colon (;) to end statements in the same way we use periods at the end of a sentence.

```
height = 32;
length = 21;
```

NAMING CONVENTIONS

- Every control class object (even the form) must begin with the appropriate 3 letter prefix. The prefix is all smalls and each word starts with a capital. There are no spaces or underscores. frmHelloWorld

- The name must represent what the variable is holding (no i or x). It must begin with a small letter and each subsequent word start with a capital. There are no spaces or underscores.

  numberOfMarks

- Constants are in all capital letters. Words are separated with underscores. The constant name must represent what the constant is holding.

  SPEED_OF_LIGHT

- Methods or Functions should begin with a capital letter and every subsequent word should also begin with a capital letter. There are no underscores. The name should indicate an action.

  PromptUser ()