Comments
```
// For single line comment
/* For multi line comment */
```

Loops
```
for (int i = 0; i < 10; i++){} //For loop
while(condition){} //while loop
foreach (var item in collection) {} //Foreach loop
do {} while (condition); //Do while loop (code runs once
for sure)
```

Conditions + switch + ternary
```
string result = (x > 0) ? "Positive" : "Non-positive"; //ternary operator

if (x > 0) { ... }
else if (x == 0) { ... }
else { ... }

switch (day) //Switch statement (with fall throgh)
{
    case "Mon":
        Console.WriteLine("Start");
        break;
    case "Fri":
        Console.WriteLine("Weekend soon");
        break;
    case “Sat”:
    case “Sun”:
        Console.WriteLine(“Weekend!! :)”);
        break;
    default:
        Console.WriteLine("Midweek");
        break;
}
```

Type conversion
```
//Implicit conversion (all information preserved)
int x = 10;
double y = x;

//Explicit conversion / cast (some information lost)
double e = 2.718281828;
int newE = (int) e; //gives 2 (not rounded up)

//Parsing strings (unsafe)
int n = int.Parse(“42”);
double d  = double.Parse(“3.14159”);

// Tryparse (safe)
if (int.TryParse(“123”, out int value)){ //Only if conversion was succesful
    Console.WriteLine(value);
}
```

Lists
```
List<int> nums = new List<int>() {1,2,3};

nums.Add(4); //Adds four to the back of the list
nums.Remove(2); //Removes the first occurrence of 2 from the list
nums.Contains(3); //Returns true if the list contains an element with the value 3
nums.Count; // The number of elements in the list
nums.Clear(); // Makes the list empty

//LINQ
nums.Where(num => num>2).ToList();
nums.Select(num => num+1).ToList();
nums.Sum();
nums.Average();
nums.Max();
nums.Min();
```

Math functions
```
using Math;
Math.Abs(-5); // 5
Math.Pow(2, 3); //8
Math.Sqrt(16); ///4
Math.Round(3.14159,2); //3.14
Math.Max(5,10); //10
Math.Min(5,10); //5
Math.PI; //3.14159265
Math.E; //2.718
```

Exception handling
```
try
{
    int x = int.Parse("oops");
}
catch (FormatException e)
{
    Console.WriteLine("Invalid input");
}
finally
{
    Console.WriteLine("Always runs");
}
```

Null handling
```
// Null handling
string? name = null;
Console.WriteLine(name ?? "Unknown");  // Null coalescing
```

Comparison
```
// Numbers
==, !=, >, <, >=, <=

// Strings
"abc" == "abc"        // true
"abc".Equals("ABC")   // false
"abc".Equals("ABC", StringComparison.OrdinalIgnoreCase)  // true

//Objects
obj1 == obj2; //Checks if they are the same object
obj1.Equals(obj2); //Checks if they contain the same values
```

Strings and chars
```
string s = "Hello";
char c = s[1];        // 'e'
foreach (char ch in s) { ... }

// Strings are immutable (new string created on modification)
string t = s.ToUpper();

// Chars are 16-bit Unicode (UTF-16)
int code = (int)'A';  // 65
char ch2 = (char)66;  // 'B'

// String interpolation
int age = 25;
Console.WriteLine($"I am {age} years old");

// Formatting
Console.WriteLine($"{Math.PI:F2}");  // 3.14
```

Files
```
// Check if file exists
if (File.Exists("data.txt")) { ... }

// Reading entire file
string text = File.ReadAllText("file.txt");

// Reading all lines
string[] lines = File.ReadAllLines("file.txt");

// Writing (overwrites)
File.WriteAllText("file.txt", "Hello");

// Writing multiple lines
File.WriteAllLines("file.txt", new string[] { "A", "B", "C" });

// Iterating through lines
foreach (string line in File.ReadLines("file.txt"))
{
    Console.WriteLine(line);
}

// Append
File.AppendAllText("output.txt", "\nMore text");
```

```
List<string> TopFiveWords()
{
    List<string> lines = File.ReadAllLines("stations.txt").ToList();

    return lines.SelectMany(line => line.Split(",")[0].Split(" ")).Select(station => station.Trim()).GroupBy(word => word).OrderByDescending(group => group.Count()).Select(group => group.Key).ToList().GetRange(0, 5);
}
```

using