

While

```
while (!isValid)
{
    Console.WriteLine( " Enter score{0}:", i + 1);
    inputValue = Console.ReadLine();
    isValid = int.TryParse (in Value, out score[i]);
    if (!isValid)
    {
        Console.WriteLine( " Invalid data entered -" + "0 stored in array");
    }
}
```

Arrays

Array Declare `int[] scores = new int[14];`

Array Initialize `int[] anArray = { 100, 200, 400, 600 };`

Copy `Copy (System.Array sourceArray, int index1Source, System.Array targetArray, int index1Target, int length-ToCopy)`

Read User Input `inValue = Console.ReadLine();`

Advanced Collections

Two dimensional Arrays `int[,] calories = { { 900, 750, 1020 }, { 300, 100, 2700 }, { 200,300,5500 } };`

Two dimensional Arrays Init `int[,] calories = new int[7,3];`

ArrayList `ArrayList anArray = new ArrayList( ); // Instantiates ArrayList`

List `List<Item> items = new List<Item>()`

Two dimensional Arrays Example

```
public static double[] CalculateAverageByDay(int[,] calories) {
    int sum = 0;
    double[] dailyAverage = new double[7];
    // 7
    for (int r = 0; r < calories.GetLength(0); r++) {
        // 3
        for (int c = 0; c < calories.GetLength(1); c++) {
            sum += calories[r, c];
        }
        dailyAverage[r] = (double) sum / calories.GetLength(1);
        sum = 0;
    }
    return dailyAverage;
}
```

Programming Based on Events

Delegates `delegate string ReturnsSimpleString( );`

Event Handler Methods `this.button1.Click += new System.EventHandlerr(this.button1_Click);`

Object Oriented Language Features

Abstraction Generalizing, identifying essential features, hiding nonessential complexities

Encapsulation Packaging data and behaviors into a single unit, hiding implementation details

Inheritance Extending program units to enable reuse of code

Polymorphism Providing multiple (different) implementations of same named behaviors

sally sung

By **sally sung**
cheatography.com/sally-sung/

Not published yet.
Last updated 12th December, 2025.
Page 1 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>

Component based applications

Business layer	Classes that perform processing / Provide functionality for specific system; provide necessary processing
Data access layer	Classes for accessing data from text files and database access
Presentation layer	User interface tier for user interaction / Graphical User Interface such as Windows or Web

Inheritance

Associated with an "is a" relationship

Classes can also have a "has a" relationship, not associated with inheritance

"has a" relationship is associated with containment or aggregation (집합)

Every object inherits four methods (Equals / GetHashCode / GetType / ToString)

Overriding Methods

```
// In order to be an method that can be
// overridden, keyword "virtual", "abstract", or
// "override" must be part of the heading for the
// parent class
public override string ToString() {
    return firstName + " " + lastName;
}
```

Creating Derived classes

```
public class Student: Person {
    private string major;
    private string studentId;
    public Student(string id, string lName, string fName, int age, string major, string sId) : base(id, lName, fName, age) {
    }
}
```

Abstract

Classes	public abstract class Example {}
Methods	public abstract string Method();

Generics

```
// Reduce the need to rewrite algorithms for each
// data type
public class GenericClass<T>
{
    public T dataMember;
}

GenericClass<string> anIdentifier = new
GenericClass<string>();
```

Working with Files

Import	using System.IO;
Exist	File.Exists(fileName)
Enumeration	public enum DayOfWeek { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday } Console.WriteLine("Today is {0}!", DayOfWeek.Tuesday);
DirectoryInfo	DirectoryInfo dir = new DirectoryInfo(".");
StreamWriter	StreamWriter outputFile = new StreamWriter("saying.txt"); outputFile.WriteLine("This is the first line in a text file");
StreamReader	StreamReader inputFile = new StreamReader("someInputFileName"); string inputValue = inputFile.ReadLine();

Inheritance - Access Modifiers

public	Accessible everywhere
private	Accessible only within the class
protected	Accessible within the class and derived classes

sally sung

By [sally sung](#)
cheatography.com/sally-sung/

Not published yet.
Last updated 12th December, 2025.
Page 2 of 2.

Sponsored by [Readable.com](#)
Measure your website readability!
<https://readable.com>