While

```
while (!isValid)
{
        Con sol e.W rit eLi ne( " Enter score{0}: ", i + 1);
        inValue = Consol e.R ead Line();
        isValid = int.Tr yPa rse (in Value, out score[i]);
       if (!isValid)
       {
                Con sol e.W rit eLi ne( " 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( ); // Instan-tiates ArrayList
List	List<Item> items = new List<Item>()

Two dimensional Arrays Example

```
public static double[] CalulateAverageByDay(int[,] calories) {
       int sum = 0;
        dou ble[] dailyA verage = new double[7];
       // 7
       for (int r = 0; r < calori es.G et Len - gth(0); r++) {
               // 3
               for (int c = 0; c < calori es.G - et Len gth(1); c++) {
                       sum += calori es[r, c];
               }
                dai lyA ver age[r] = (double) sum / calori es.G et Len gth(1);
               sum = 0;
       }
        return dailyA verage;
}
```

Programming Based on Events

Delegates	delegate string ReturnsSimpleString( );
Event Handler Methods	this.button1.Click += new System.EventHandle-r(this.button1_Click);

Object Oriented Language Features

Abstra-ction	Generalizing, identifying essential features, hiding nonessential complexities
Encaps-ulation	Packaging data and behaviors into a single unit, hiding implementation details
Inheri-tance	Extending program units to enable reuse of code
Polymo-rphism	Providing multiple (different) implementations of same named behaviors

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
Presen-tation 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 Studen t(s tring 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 Generi cCl ass <T>
{
public T dataMe mber;
}
Generi cClass <st rin g> anIden tifier = new Generi cCl ass <st rin g>();
```

Working with Files

Import	using System.IO;
Exist	File.Exists(fileName)
Enumer ation	public enum DayOfWeek { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday } Console.Writ-eLine("Today is {0}!", DayOfWeek.Tuesday);
Direct-oryInfo	DirectoryInfo dir = new DirectoryInfo(".");
Stream Writer	StreamWriter outputFile = new StreamWriter("saying.txt"); outputFile.WriteLine("This is the first line in a text file");
Stream Reader	StreamReader inputFile = new StreamReader("someInpu-tFileName"); string inValue = inputFile.ReadLine();

Inheritance - Access Modifiers

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

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