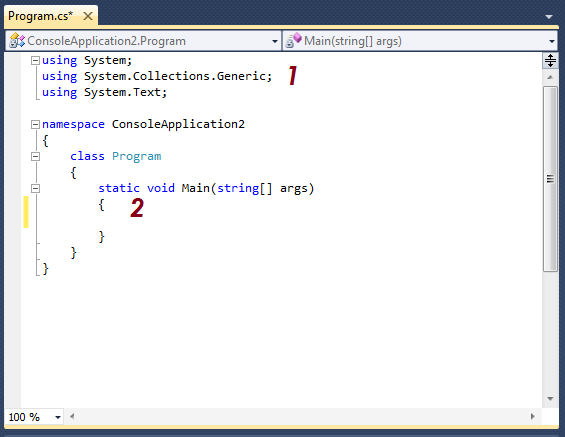
**Lecture # 2:**

**Structure of C#:**

* C# is case sensitive language
* C# is pure object oriented programming
* Below is default format of C# Console Application



1. Namespaces references
2. Main function where program execution starts.

**C# Basic Programming concepts:**

* There are two types of C# applications which are commonly developed. One is called console based application and other is called form based applications.
* We will start learning C# by designing console based applications.

**Basic Program Structure Components of a C# Program:**

**Main method:**

* Every C# application must have a method named *Main* defined in one of its classes. It doesn't matter which class contains the method—you can have as many classes as you

want in a given application—as long as one class has a method named *Main*. It is the entry point of your program, where the program control starts and ends.

* The entry point of the C# application is the Main method.
* There can only be one, and it is a static method in a class.
* The Main method can be written without parameters or with parameters. There are three ways to declare the Main method:
  1. It can return void as in public static void Main() { ... }
  2. It can also return an int as in public static int Main()

{

return 0;

}

* 1. It can also take arguments as in public static int Main(string[] args)

{

return 0;

}

* The method usually returns void and is passed command-line arguments as an array of strings.

public static void Main(string[] args)

{

}

**Namespaces:**

* A namespace (similar to a package in Java) called System that is provided by the .NET runtime. System is the root of the .NET base class namespace. using is very similar in concept to Java's import keyword.
* Unlike Java, in C# you cannot import a single class, but the whole package. However, the components of a namespace name do not have to map onto directories (remember in Java, a package has to physically map to a directory).
* Namespaces are a part of a type name and they are used to group and/or distinguish named entities from other ones.

A namespace is defined like this:

namespace FooNamespace

{

// Members

}

**Keywords:**

Because keywords have specific meaning in C#, you can't use them as identifiers for something else, such as variables, constants, class names, and so on. However, they can be used as part of a longer token, for example: public int abstract\_int; Also, because C# is case sensitive, if a programmer is bent on using one of these words as an identifier of some sort, you can use an

initial uppercase letter. While this is possible, it is a very bad idea in terms of human readability. There are numerous Classes defined in the standard packages. While their names are not keywords, the overuse of these names may make your meaning unclear to future people working on your programs. Following are some keywords used in C# language.

abstract base bool break byte case catch char checked class

const continue decimal default delegate do double else enum event explicit extern false finally fixed float for foreach goto if

implicit in int interface internal is lock long namespace new

null object operator out override params private protected public readonly ref return sbyte sealed short sizeof static string struct switch

this throw true try typeof uint ulong unchecked unsafe ushort

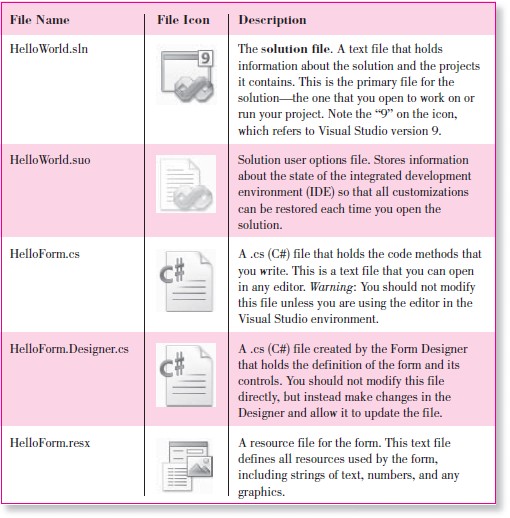
***Using statement***

* The using statement loads a specific namespace from a referenced assembly.
* It is usually placed in the top (or header) of a code file but it can be placed elsewhere if wanted, e.g. inside classes.

using System;

using System.Collections;

**Visual Studio .Net Projects / Solutions files:**



**First Application in C#:**

**using System; namespace FirstAppNS**

**{**

**class FirstApp**

**{**

**public static void Main()**

**{**

**Console.WriteLine (“First Application in C#”);**

**}**

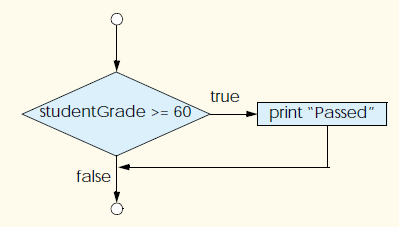
**}**

**}**

* Above application will be designed as a console application in visual studio .net.
* For debugging of this application, use F5 key from keyboard or “Start Debugging” option from “Debug” main menu.
* Result of this application will be displayed on a Console/DOS environment based window.

**Basic Programming constructs of C#:**

**Conditional Statements:** There are three types of conditional statements in C#.



1. Single way selection statements.
2. Two way selection statements.
3. Multi way selection statements.

**Single Way Selection:**

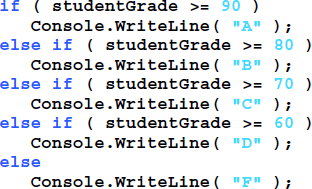
**if (condition) statement;**

**Two Way Selection:**

**if (condition)**

**statement1;**

**else**



**statement2;**

**Multi Way Selection:**

**if (condition1)**

**statement1; else if (condition2)**

**statement2; else if (condition3)**

**statement3;**

**else**

# Program#1:

**statement4;**

Write a program which took marks obtained and total marks and show grades using following grading criteria:

* If percentage <50 Show F GRADE
* If percentage is between 50 to 60 show C GRADE
* If percentage is between 60 to 70 show B GRADE
* If percentage is > 70 show A GRADE

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

namespace LabTask1

{

class Program

{

static void Main(string[] args)

{

float markobtained, totalmark; float per;

Console.WriteLine("Please enter marks obtained: "); markobtained = int.Parse(Console.ReadLine());

Console.WriteLine("Please enter total marks: "); totalmark = int.Parse(Console.ReadLine());

per = (markobtained / totalmark) \* 100; Console.WriteLine("Percentage is: " + Math.Round(per,2));

if (per > 70)

Console.Write("Your Grade is A");

else if (per >= 60 && per <= 70) Console.WriteLine("Your Grade is B");

else if (per >= 50 && per <= 60) Console.WriteLine("Your grade is C");

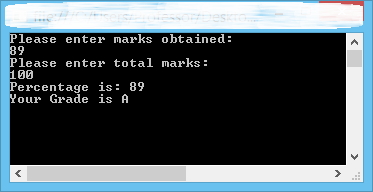
else

Console.WriteLine("Your grade is F"); Console.ReadLine();

}

}

}



# Program#2:

Write a program which took two input as integer and one as choice. Users have choice to choose one of the following operations “ + , - , \* , / , % ”. Show relevant results.

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

namespace Grades

{

class Program

{

static void Main(string[] args)

{

int var1, var2; char opert;

Console.WriteLine("Please Enter 1st value: "); var1 = int.Parse(Console.ReadLine());

Console.WriteLine("Please Enter 2nd value: "); var2 = int.Parse(Console.ReadLine());

Console.WriteLine("Please Enter select your desired operator (+, -,

/, \*, %): ");

opert = Convert.ToChar(Console.ReadLine());

try

{

switch (opert)

{

case '+':

Console.Write("The addition is equal to: "); Console.WriteLine(var1 + var2);

break;

case '-':

Console.Write("The subtarction is equal to: "); Console.WriteLine(var1 - var2);

break;

case '\*':

Console.Write("The multiplication is equal to: "); Console.WriteLine(var1 \* var2);

break;

case '%':

Console.Write("The percentage is equal to: "); Console.WriteLine(var1 % var2);

break;

case '/':

if (var2 == 0)

{

Console.WriteLine("Can not divided");

}

else

{

Console.Write("The division is equal to: "); Console.WriteLine(var1 / var2);

}

break;

default:

Console.WriteLine("You have entered wrong operator"); break;

}

}

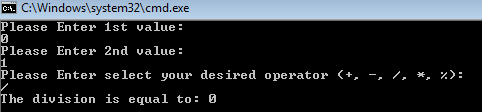
catch { }

Console.ReadLine();

}

}

}



# Program#3:

Create an array of 10 index. Input data from user and ask user to input a number to search. Show location and found message.

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

namespace Grades

{

class Program

{

static void Main(string[] args)

{

int[] arrayname = new int[10]; int searchedno;

bool result = false;

Console.WriteLine("Please enter 10 numbers: "); for (int i = 0; i < arrayname.Length; i++)

{

arrayname[i] = int.Parse(Console.ReadLine());

}

Console.WriteLine("Enter no to search: "); searchedno = int.Parse(Console.ReadLine());

for (int i = 0; i < arrayname.Length; i++)

{

if (arrayname[i] == searchedno)

{

Console.WriteLine("Found at index: " + i); result = true;

}

}

if (result == false)

{

Console.WriteLine("Number not Found");

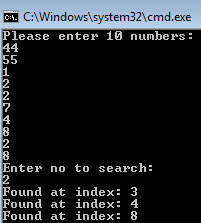
}

Console.ReadLine();

}

}

}



Program#4:

Input an array of 10 index and search largest number from the array and show along with index number.

using System;

using System.Collections.Generic;

using System.Linq; using System.Text;

namespace Grades

{

class Program

{

static void Main(string[] args)

{

int[] ArrayInt = new int[10];

for (int i = 0; i < 10; i++)

{

Console.WriteLine("Please Enter value of index: " + i); ArrayInt[i] = int.Parse(Console.ReadLine());

}

for (int i = 0; i < 10; i++)

{

Console.WriteLine("The value of " + i); Console.WriteLine(ArrayInt[i]);

}

int max = ArrayInt[0];

try

{

for (int i = 0; i < 10; i++)

{

if (max < ArrayInt[i])

{

int temp = max; max = ArrayInt[i]; ArrayInt[i] = temp;

}

}

}

catch { }

Console.WriteLine("The Maximum number in the list is: " + max);

Console.ReadLine();

}

}

}

