Objectives:

- **Data Type**
- **Operators**
- **Input and Output**
- **Conversion**

1. Create a simple C# Application. Save file as Lab1.cs
    a. Open **Start\Programs\Microsoft Visual Studio\ Microsoft Visual Studio.**
    b. Choose **File\New\Project t**o create a new **C# Console application** from **Template** list, click **Console Application.**
    c. To select **ConsoleApplication1** in the **Name** box, click **ConsoleApplication1**.
    d. Change name as **Lab1.**
    e. Type the below code.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Example
{
    class Example
    {
        static void Main(string[] args)
        {
            Console.WriteLine("hello world");
            Console.ReadLine();
        }
    }
}
```

    f. To save this file, select **File\SaveProgram.cs**

g. To compile this program select: **Build\Bulid Program.cs**

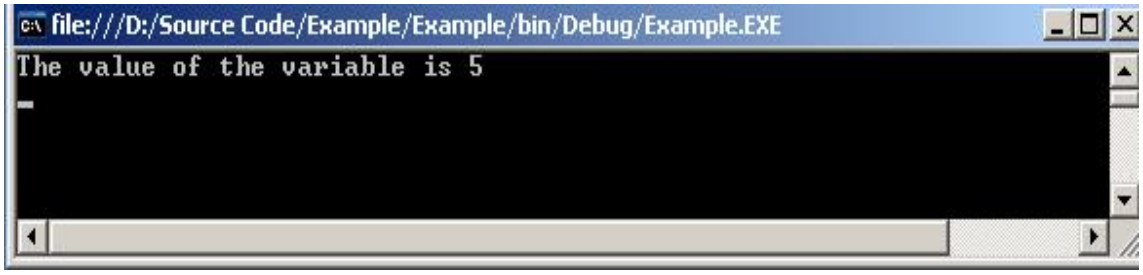
h. To execute this program select: **Debug\Start Without Debugging**

2. How to use data types and variables

A. How to use value type:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Session2
{
    class Example1
    {
        static void Main(string[] args)
        {
            int valueVal = 5;
            Test(valueVal);
            Console.WriteLine("The value of the variable is {0}",
valueVal);
            Console.ReadLine();
        }
        static void Test(int valueVal)
        {
            int temp = 5;
            valueVal = temp * 2;
        }
    }
}
```

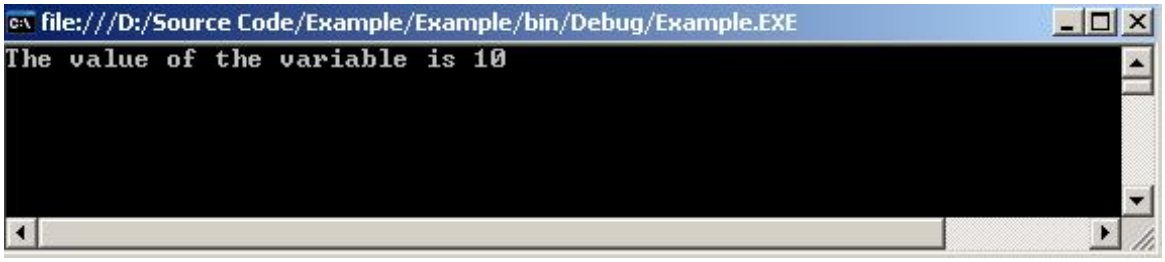
The output of the program:



B. How to use Reference Type

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Session2
{
    class ReferenceType
    {
        public int valueVal;
    }
    class Example2
    {
        static void Main(string[] args)
        {
            ReferenceType refer = new ReferenceType();
            refer.valueVal = 5;
            Test(refer);
            Console.WriteLine("The value of the variable is
{0}",refer.valueVal);
            Console.ReadLine();
        }
        static void Test(ReferenceType refer)
        {
            int temp = 5;
            refer.valueVal = temp * 2;
        }
    }
}
```

The output of the program:



3. How to use **keyword, comment, constant and literals**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Session
{

    class Example3
    {
        static void Main(string[] args)
        {
            /*
             Comment make the program more readable.
             They help the programer to explain the purpose of  using a
particular or method.
             Comment are ignored by the compiler during the execution
of the program.
             There are three kind of comments such as: Single-line
comment, Multi-line comment, XML comment.
             */
            //this is single-line comment
            //assigned constant and literal variable
            const int age = 25;
            //Boolean literal assigned to the variable
```

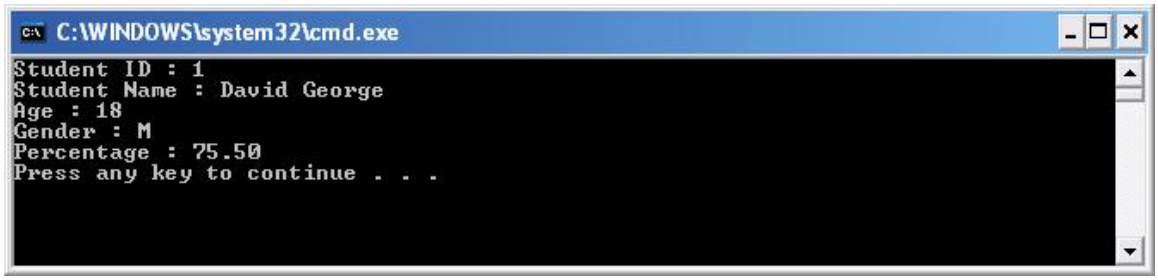
```
            bool gender = true;//true is male, false is female
            //Integer literal can be assigned to int,uint,long,ulong
datatypes
            long IDNumber = 123456;
            //Real literal canbe assigned to float,double
            float mark = 8.5f;
            //Character literal can be assigned to a char  data type
            char blood = 'A';
            //String literal can be assigned two types string :regular
and verbatim
            //regular:
            string name = "Tran The Hai";
            //verbatim
            string email = "@yahoo.com";
            //Null literal has only one value,null:string name=null;
            string status = null;
            //We can use keyword to named variable
            string @string = "FPT-APTECH";

            Console.WriteLine("The information detail of a person:");
            Console.WriteLine("Fullname:{0}\n",name);//\n is a escape
sequence
            Console.WriteLine("Age:{0}\t",age);
            Console.WriteLine("Gender:{0}\n",gender);
            Console.WriteLine("IDNumber:{0}\n",IDNumber);
            Console.WriteLine("Blood Group:{0}\n",blood);
            Console.WriteLine("Email:{0}\n",email);
            Console.WriteLine("Status:{0}\n\t",status);
            Console.WriteLine("Company:{0}",@string);
            Console.ReadLine();
        }

    }
}
```

4. How to use input and output

A. Type this code:

```
/*
*Students.cs
*/
using System;
using System.Collections.Generic;
using System.Text;

namespace School
{
    /* The program creates an Student class to declare,
     intialise and display the variables.Student class stores and
displays the details of students
     using different data types.
     */
    class Student
    {

        static void Main(string[] args)
        {
        //Declaring and intialising variables to store student details
            int id = 1;
            string name = "David George";
            byte age = 18;
            char gender = 'M';
            float percent = 75.50F;

            //Displaying the student details
            Console.WriteLine("Student ID : {0}", id);
            Console.WriteLine("Student Name : {0}", name);
            Console.WriteLine("Age : " + age);
            Console.WriteLine("Gender : " + gender);
            Console.WriteLine("Percentage : {0:F2}", percent);
        }
    }
}
```

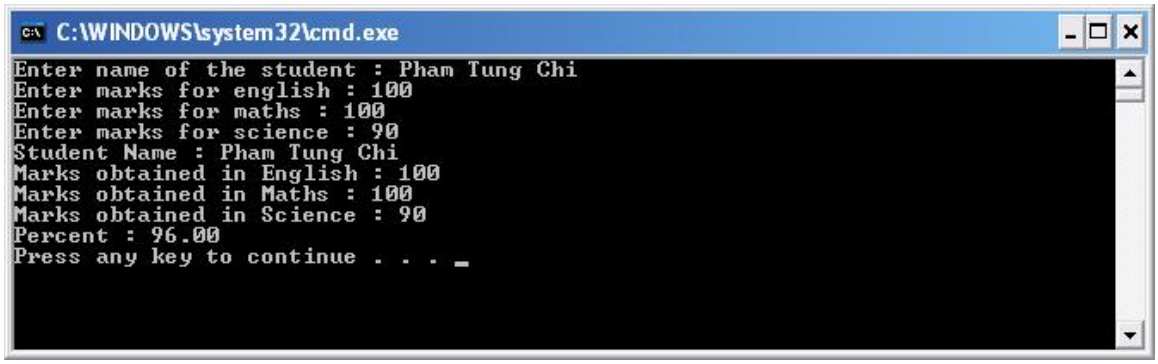
The output of the program:



B. Type this below code:

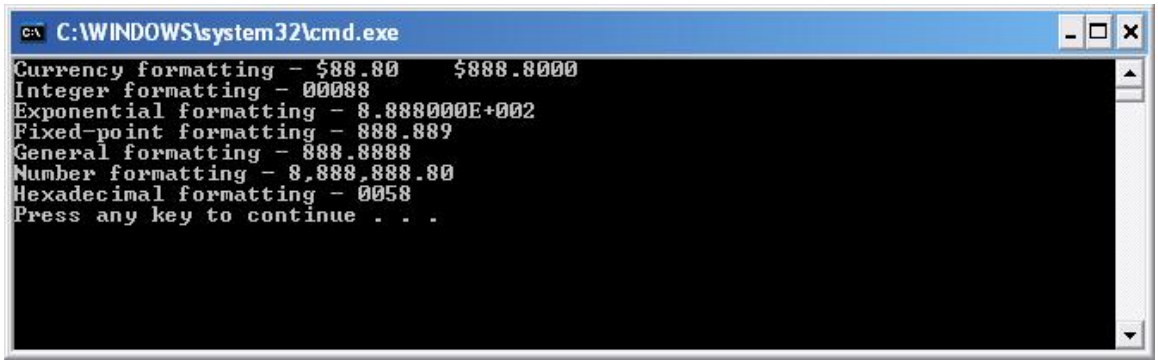
```
using System;
using System.Collections.Generic;
using System.Text;

      /* The program demonstrates the input and output operations.*/

class Student
      {
            static void Main(string[] args)
            {
                  //Declaring integer constant to store value 100
                  const int percentConst = 100;

                  //Declaring variable to store the student name
                  string studentName;

                  //Declaring variables to store the student marks
                  int english, maths, science;

      //Declaring and initialising variable to store the percentage
                  float percent = 0.0F;

                  //Accepting the details of the student
                  Console.Write("Enter name of the student : ");
```

```
                studentName = Console.ReadLine();

                Console.Write("Enter marks for english : ");
                english = Convert.ToInt32(Console.ReadLine());

                Console.Write("Enter marks for maths : ");
                maths = Convert.ToInt32(Console.ReadLine());

                Console.Write("Enter marks for science : ");
                science = Convert.ToInt32(Console.ReadLine());

                //Calculating the percentage of the student
      percent = ((english + maths + science) * percentConst) / 300;

                //Displaying the details of the student
                Console.WriteLine("Student Name : " + studentName);

                Console.WriteLine("Marks obtained in English : {0}",
english);
                Console.WriteLine("Marks obtained in Maths : {0}",
maths);
                Console.WriteLine("Marks obtained in Science : {0}",
science);

                Console.WriteLine("Percent : {0:F2}", percent);
            }
      }
```



5. Numeric formatting

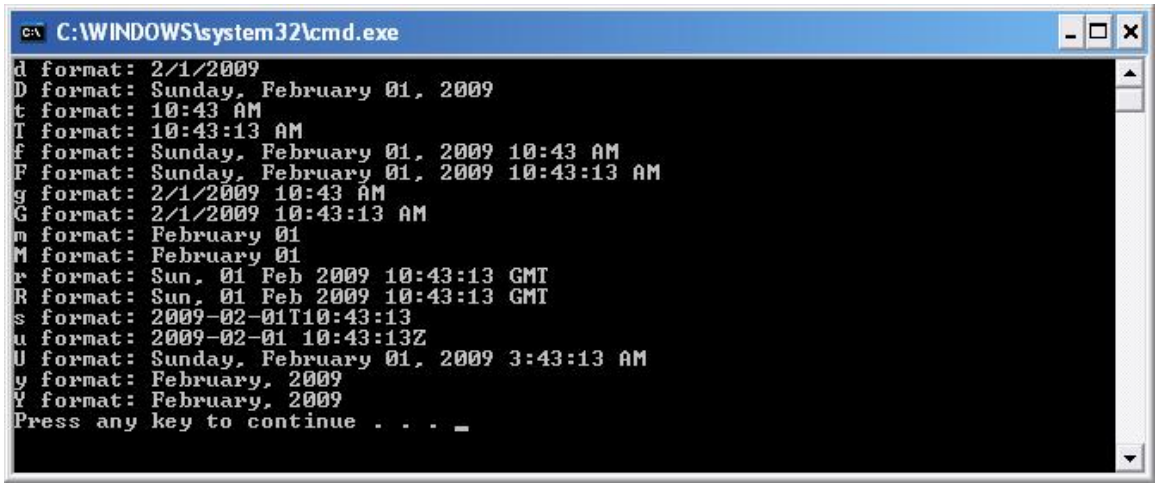
```
/*
 * NumFormat.cs
 */

using System;
using System.Collections.Generic;
using System.Text;

class NumberFormat
{
    static void Main(string[] args)
    {
        Console.WriteLine("Currency formatting - {0:C}      {1:C4}",
88.8, 888.8);
        Console.WriteLine("Integer formatting - {0:D5}", 88);
        Console.WriteLine("Exponential formatting - {0:E}", 888.8);
        Console.WriteLine("Fixed-point formatting - {0:F3}", 888.8888);
        Console.WriteLine("General formatting - {0:G}", 888.8888);
        Console.WriteLine("Number formatting - {0:N}", 8888888.8);
        Console.WriteLine("Hexadecimal formatting - {0:X4}", 88);
    }
}// End of class
```



6. Datetime formatting

a. Example 1

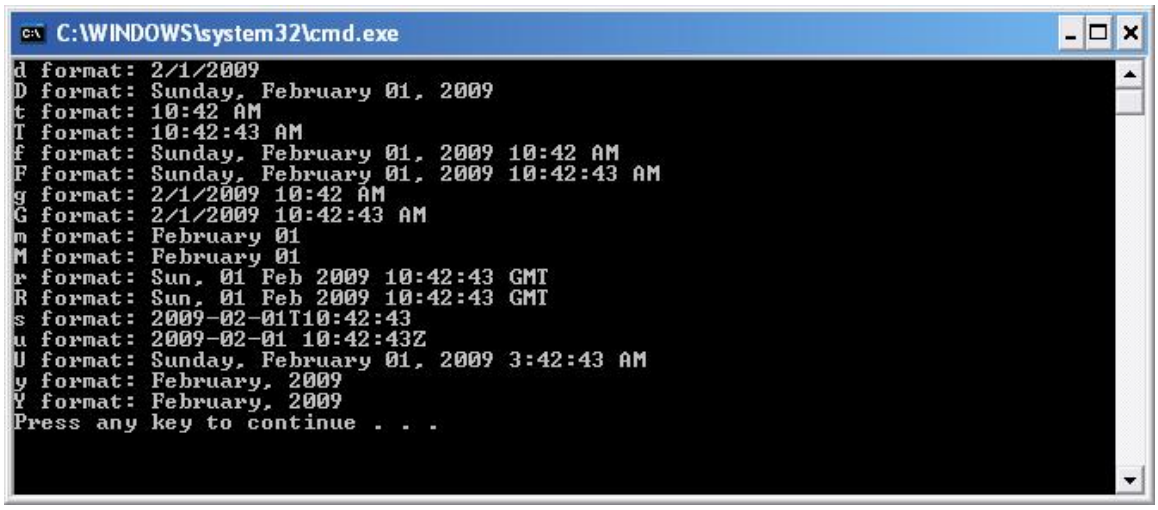
```
using System;

class MainClass
{
    public static void Main()
    {
        DateTime dt = DateTime.Now;

        Console.WriteLine("Time is {0:hh:mm tt}", dt);
        Console.WriteLine("24 hour time is {0:HH:mm}", dt);
        Console.WriteLine("Date is {0:ddd MMM dd, yyyy}", dt);
        Console.WriteLine("Time with seconds: " + "{0:HH:mm:ss tt}",
dt);
        Console.WriteLine("Use m for day of month: {0:m}", dt);
        Console.WriteLine("use m for minutes: {0:%m}", dt);
    }
}
```



b. Example 2

```
using System;
class MainClass
{
    public static void Main()
    {
        DateTime dt = DateTime.Now; // obtain current time

        Console.WriteLine("d format: {0:d}", dt);
        Console.WriteLine("D format: {0:D}", dt);

        Console.WriteLine("t format: {0:t}", dt);
        Console.WriteLine("T format: {0:T}", dt);

        Console.WriteLine("f format: {0:f}", dt);
        Console.WriteLine("F format: {0:F}", dt);

        Console.WriteLine("g format: {0:g}", dt);
        Console.WriteLine("G format: {0:G}", dt);

        Console.WriteLine("m format: {0:m}", dt);
        Console.WriteLine("M format: {0:M}", dt);

        Console.WriteLine("r format: {0:r}", dt);
        Console.WriteLine("R format: {0:R}", dt);
```

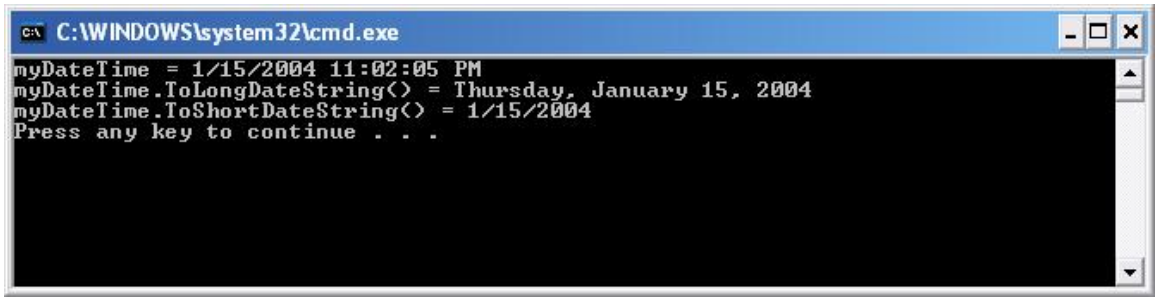
```
        Console.WriteLine("s format: {0:s}", dt);

        Console.WriteLine("u format: {0:u}", dt);
        Console.WriteLine("U format: {0:U}", dt);

        Console.WriteLine("y format: {0:y}", dt);
        Console.WriteLine("Y format: {0:Y}", dt);
    }
}
```



c. Example 3

```
using System;

class MainClass
{
    public static void Main()
    {
        DateTime myDateTime = new DateTime(2004, 1, 15, 23, 2, 5);
        Console.WriteLine("myDateTime = " + myDateTime);
        Console.WriteLine("myDateTime.ToLongDateString() = " +
myDateTime.ToLongDateString());
        Console.WriteLine("myDateTime.ToShortDateString() = " +
myDateTime.ToShortDateString());
```

```
    }
}
```



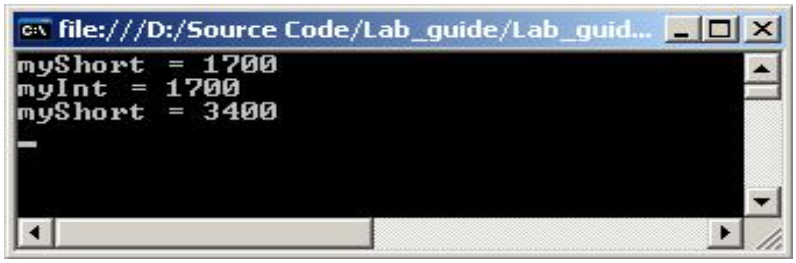
7. Conversion

a. Type below snippet code:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab_guide
{
    class ConvertDemo1
    {
        static void Main(string[] args)
        {
            short myShort = 1700;
            System.Console.WriteLine("myShort = " + myShort);

            int myInt = myShort;
            System.Console.WriteLine("myInt = " + myInt);

            myShort = (short)(myInt * 2);
            System.Console.WriteLine("myShort = " + myShort);
            Console.ReadLine();

        }
    }
```

```
}
```

The output of the program:



b. This snippet code below explains how to clearly implicit and explicit convert datatype.

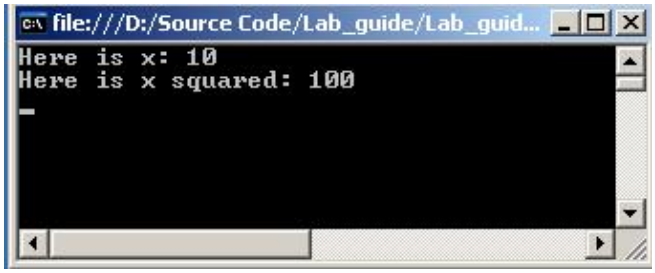
```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab_guide
{
    class ConvertDemo2
    {
        static void Main(string[] args)
        {
            // all implicit
            sbyte v = 55;
            short v2 = v;
            int v3 = v2;
            long v4 = v3;

            // explicit to "smaller" types
            v3 = (int) v4;
            v2 = (short) v3;
            v = (sbyte) v2;
            Console.WriteLine("v="+v);
            Console.WriteLine("v2="+v2);
            Console.WriteLine("v3="+v3);
            Console.WriteLine("v4="+v4);
```

```
            Console.ReadLine();
        }
    }
}
```

c. This below code that boxing occurs when passing values. Type it now.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab_guide
{
    class ConvertDemo4
    {
        static void Main(string[] args)
        {
            int x;
            x = 10;

            Console.WriteLine("Here is x: " + x);

            // x is automatically boxed when passed to sqr()
            x = ConvertDemo4.boxObject(x);
            Console.WriteLine("Here is x squared: " + x);
            Console.ReadLine();
        }
        static int boxObject(object o)
        {
            return (int)o * (int)o;
        }

    }
}
```

The output of the program:

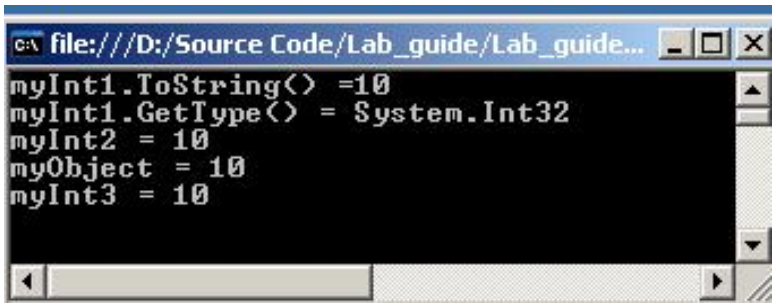


d. This below code that explicit and implicit boxing of an integer. Type it now.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab_guide
{
    class ConvertDemo5
    {
        static void Main(string[] args)
        {
            // implicit boxing of an int
            int myInt1 = 10;
            Console.WriteLine("myInt1.ToString() =" +
myInt1.ToString());
            Console.WriteLine("myInt1.GetType() = " +
myInt1.GetType());

            // explicit boxing of an int to an object
            int myInt2 = 10;
            object myObject = myInt2;  // myInt2 is boxed
            Console.WriteLine("myInt2 = " + myInt2);
            Console.WriteLine("myObject = " + myObject);

            // explicit unboxing of an object to an int
            int myInt3 = (int)myObject;  // myObject is unboxed
            Console.WriteLine("myInt3 = " + myInt3);
```

```
            Console.ReadLine();

        }

    }
}
```

The output of the program:



Do It Yourself

1.1. Write a program to enter: name, address, phone and display these information.

1.2. Write a program to calculate average of subjects include: math, physical, chemical, foreign language, history.

1.3. Do Workshop 1, 2 in CD.

1.4. Do ACTCSharp_Module1_Assignment.pdf in CD.

1.5. Do ACTCSharp_Module2_Assignment.pdf in CD.

References

1) **CD ROM C# Programming, Aptech Education**
2) **http://www.java2s.com/Tutorial/CSharp/CatalogCSharp.htm**
3) **MSDN Document**
4) **[ebook] MSDN training, Introduction to C#, Microsoft Press**