



Demo document

Phase - I



Prepared By:
Pinal Pambhar

Table of Contents

Windows App, Class Library-----	1
Class Library-----	1
Windows Application -----	2
Create First C# Program “Hello World”-----	6
What is Namespace? -----	6
What is Class? -----	7
Variable And Method Declaration-----	8
Understanding datatypes & variables with conversion -----	9
Datatype Conversion -----	9
Implicit / Automatic Type Conversion:-----	9
Explicit Type conversion: -----	10
Boxing/Unboxing -----	11
Boxing-----	11
Unboxing -----	12
Understanding Decision making & statements-----	13
if-else -----	13
if-else statement -----	13
else-if statement -----	14
Switch -----	15

Windows App, Class Library

Class Library

Calculator.cs

```
using System;

namespace ClassLibrary_Calculator
{
    public class Calculator
    {
        #region Method_Addition
        public double Addition(double Number1 , double Number2)
        {
            return Number1 + Number2;
        }
        #endregion Method_Addition

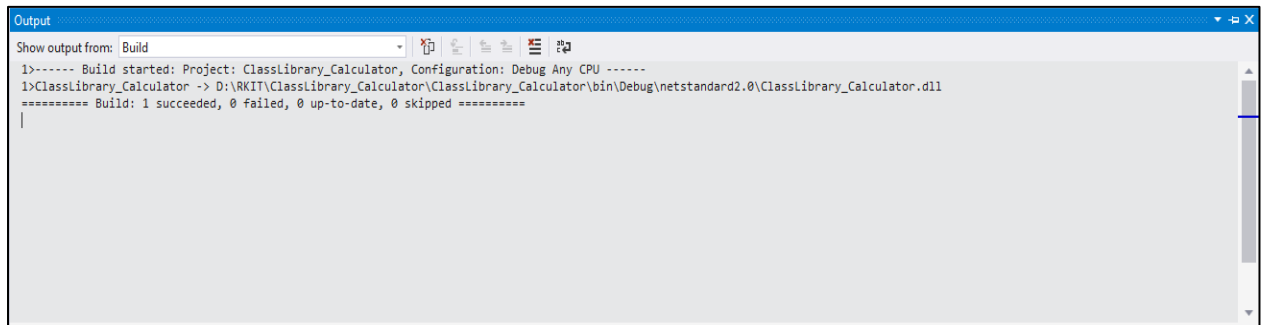
        #region Method_Subtraction
        public double Subtraction(double Number1, double Number2)
        {
            return Number1 - Number2;
        }
        #endregion Method_Subtraction

        #region Method_Multiplication
        public double Multiplication(double Number1, double Number2)
        {
            return Number1 * Number2;
        }
        #endregion Method_Multiplication

        #region Method_Division
        public double Division(double Number1, double Number2)
        {
            return Number1 / Number2;
        }
        #endregion Method_Division

        #region Method_Modulo
        public double Modulo(double Number1, double Number2)
        {
            return Number1 % Number2;
        }
        #endregion Method_Modulo
    }
}
```

Output:



Windows Application

- Here Windows Application (Calculator) is using Class Library (Calculator.cs).

FormCalculator.cs

```
using System;
using System.Windows.Forms;
using ClassLibrary_Calculator;

namespace WindowsFormsApp_Calculator
{
    public partial class FormCalculator : Form
    {
        #region Constructor
        public FormCalculator()
        {
            InitializeComponent();
        }
        #endregion Constructor

        #region objects_declaration
        Calculator objcalculator = new Calculator();
        #endregion objects_declaration

        #region method_btnAddition
        private void btnAddition_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(txtNumber1.Text) || string.IsNullOrEmpty(
txtNumber2.Text))
            {
                txtAnswer.Text = "Number can't be blank";
            }
            else if (Convert.ToDouble(txtNumber1.Text) < 0 || Convert.ToDouble
(txtNumber2.Text) < 0)
            {
                txtAnswer.Text = "Invalid Input";
            }
            else
            {
                double answer;
                lblAnswer.Text = "(" + txtNumber1.Text.ToString().Trim() + " + " +
txtNumber2.Text.ToString().Trim() + ")";
            }
        }
    }
}
```

```

        answer = objcalculator.Addition(Convert.ToDouble(txtNumber1.Text),
        Convert.ToDouble(txtNumber2.Text));

        txtAnswer.Text = Convert.ToString(answer);
    }
}
#endregion method_btnAddition

#region method_btnSubtraction
private void btnSubtraction_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtNumber1.Text) || string.IsNullOrEmpty(
txtNumber2.Text))
    {
        txtAnswer.Text = "Number can't be blank";
    }
    else if (Convert.ToDouble(txtNumber1.Text) < 0 || Convert.ToDouble
(txtNumber2.Text) < 0)
    {
        txtAnswer.Text = "Invalid Input";
    }
    else
    {
        double answer;
        lblAnswer.Text = "(" + txtNumber1.Text.ToString().Trim() + " -
" + txtNumber2.Text.ToString().Trim() + ")";

        answer = objcalculator.Subtraction(Convert.ToDouble(txtNumber1.Text),
Convert.ToDouble(txtNumber2.Text));

        txtAnswer.Text = Convert.ToString(answer);
    }
}
#endregion method_btnSubtraction

#region method_btnClear
private void btnClear_Click(object sender, EventArgs e)
{
    txtNumber1.Text = String.Empty;
    txtNumber2.Text = String.Empty;
    txtAnswer.Text = String.Empty;
    lblAnswer.Text = String.Empty;
}
#endregion method_btnClear

#region method_btnMultiplication
private void btnMultiplication_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtNumber1.Text) || string.IsNullOrEmpty(
txtNumber2.Text))
    {
        txtAnswer.Text = "Number can't be blank";
    }
    else if (Convert.ToDouble(txtNumber1.Text) < 0 || Convert.ToDouble
(txtNumber2.Text) < 0)
    {
        txtAnswer.Text = "Invalid Input";
    }
}
}

```

```

        else
        {
            double answer;
            lblAnswer.Text = "(" + txtNumber1.Text.ToString().Trim() + " * " +
            txtNumber2.Text.ToString().Trim() + ")";

            answer = objcalculator.Multiplication(Convert.ToDouble(
            txtNumber1.Text), Convert.ToDouble(txtNumber2.Text));

            txtAnswer.Text = Convert.ToString(answer);
        }
    }
#endregion method_btnMultiplication

#region method_btnDivision
private void btnDivision_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtNumber1.Text) || string.IsNullOrEmpty(
    txtNumber2.Text))
    {
        txtAnswer.Text = "Number can't be blank";
    }
    else if (Convert.ToDouble(txtNumber1.Text) < 0 || Convert.ToDouble(
    txtNumber2.Text) < 0)
    {
        txtAnswer.Text = "Invalid Input";
    }
    else
    {
        double answer;
        lblAnswer.Text = "(" + txtNumber1.Text.ToString().Trim() + " / " +
        txtNumber2.Text.ToString().Trim() + ")";

        answer = objcalculator.Division(Convert.ToDouble(txtNumber1.Text),
        Convert.ToDouble(txtNumber2.Text));

        txtAnswer.Text = Convert.ToString(answer);
    }
}
#endregion method_btnDivision

#region method_btnModulo
private void btnModulo_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtNumber1.Text) || string.IsNullOrEmpty(
    txtNumber2.Text))
    {
        txtAnswer.Text = "Number can't be blank";
    }
    else if (Convert.ToDouble(txtNumber1.Text) < 0 || Convert.ToDouble(
    txtNumber2.Text) < 0)
    {
        txtAnswer.Text = "Invalid Input";
    }
    else
    {
        double answer;
        lblAnswer.Text = "(" + txtNumber1.Text.ToString().Trim() + " % " +
        txtNumber2.Text.ToString().Trim() + ")";
    }
}
#endregion method_btnModulo

```

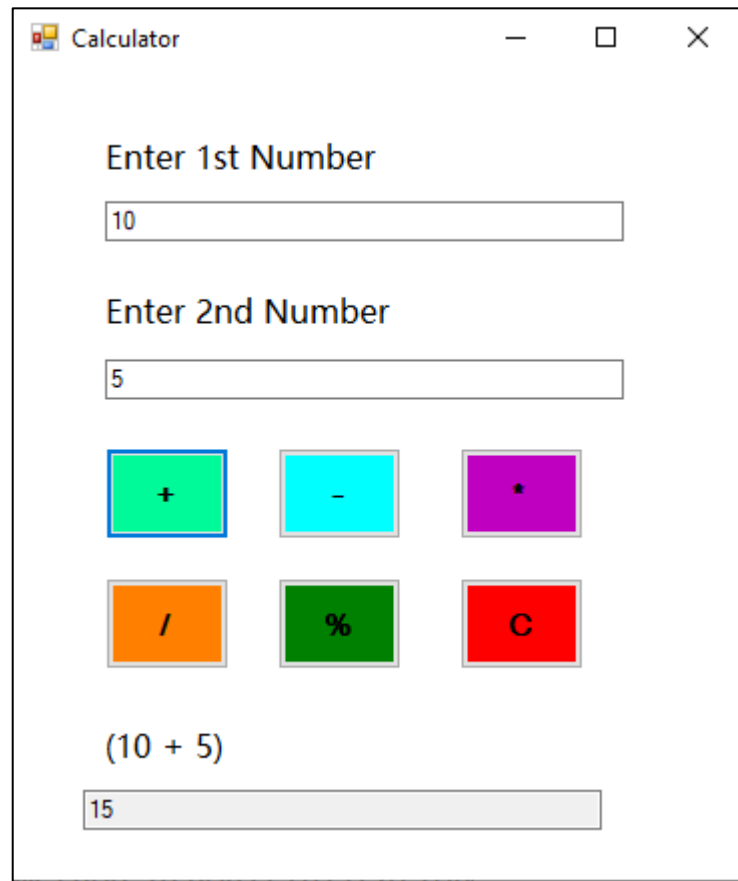
```

        answer = objcalculator.Modulo(Convert.ToDouble(txtNumber1.Text),
        Convert.ToDouble(txtNumber2.Text));

        txtAnswer.Text = Convert.ToString(answer);
    }
}
#endregion method_btnModulo
}

```

Output:



Create First C# Program “Hello World”

What is Namespace?

DemoNamespace.cs

```
using System;

namespace MyNewNamespace
{
    class DemoNamespace
    {
        public static void MyMethod()
        {
            System.Console.WriteLine("Creating My Namespace");
        }

        public void ILoveNamespace()
        {
            System.Console.WriteLine("I Love Namespace");
        }
    }
}
```

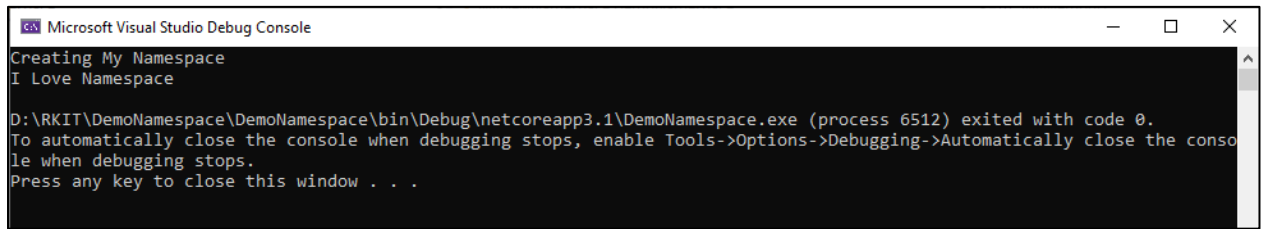
Using_Namespace.cs

```
using System;
using MyNewNamespace;

namespace Using_DemoNamespace
{
    class Using_Namespace
    {
        static void Main(string[] args)
        {
            // Method-1 to use namespace methods
            MyNewNamespace.DemoNamespace.MyMethod();

            //Method-2 to use namespace methods
            DemoNamespace objNamespace = new DemoNamespace();
            objNamespace.ILoveNamespace();
        }
    }
}
```


Output:



```
Microsoft Visual Studio Debug Console
Creating My Namespace
I Love Namespace

D:\RKIT\DemoNamespace\DemoNamespace\bin\Debug\netcoreapp3.1\DemoNamespace.exe (process 6512) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

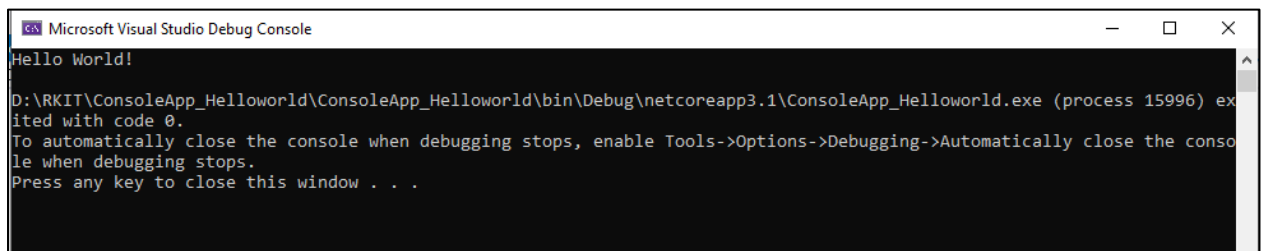
What is Class?

Helloworld.cs

```
using System;

namespace ConsoleApp_Helloworld
{
    class Helloworld
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Output:



```
Microsoft Visual Studio Debug Console
Hello World!

D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 15996) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Variable And Method Declaration

Helloworld.cs

```
using System;

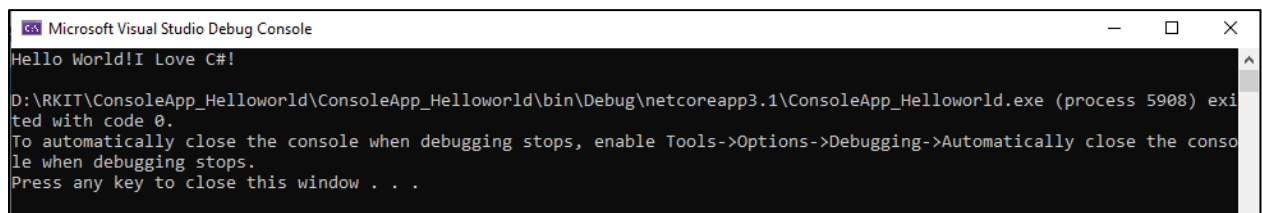
namespace ConsoleApp_Helloworld
{
    class Helloworld
    {
        static void Main(string[] args)
        {
            //declaring string variable newword
            string newwords;

            //Assigning value to the variable newword
            newwords = "I Love C#!";

            //declaring a method printConsole
            void printConsole()
            {
                Console.WriteLine("Hello World!" + newwords);
            }

            //calling method printConsole
            printConsole();
        }
    }
}
```

Output:



The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The console output is as follows:

```
Hello World!I Love C#!

D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 5908) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Understanding datatypes & variables with conversion

Datatype Conversion

Implicit / Automatic Type Conversion:

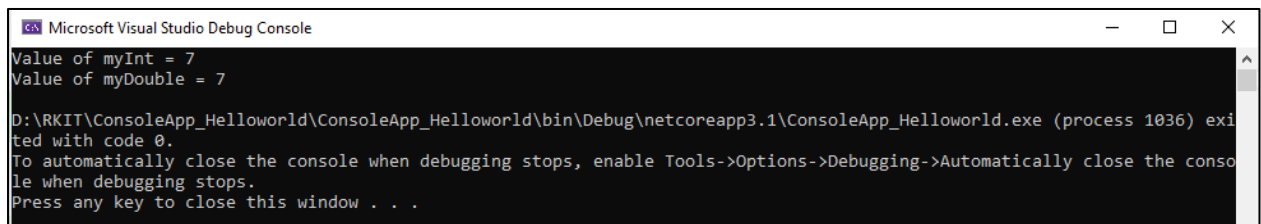
Datatype_conversion.cs

```
using System;

namespace ConsoleApp_Helloworld
{
    class Datatype_conversion
    {
        static void Main(string[] args)
        {
            int myInt = 7;
            double myDouble = myInt;    // Implicit casting: int to double

            Console.WriteLine("Value of myInt = " + myInt);
            Console.WriteLine("Value of myDouble = " + myDouble);
        }
    }
}
```

Output:



The screenshot shows the Microsoft Visual Studio Debug Console window. The output text is as follows:

```
Value of myInt = 7
Value of myDouble = 7

D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 1036) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Explicit Type conversion:

Datatype_conversion.cs

```
using System;

namespace ConsoleApp_Helloworld
{
    class Datatype_conversion
    {
        static void Main(string[] args)
        {
            double myDouble = 9.78;
            int myInt = (int)myDouble;    // Explicit casting manually:
            double to int

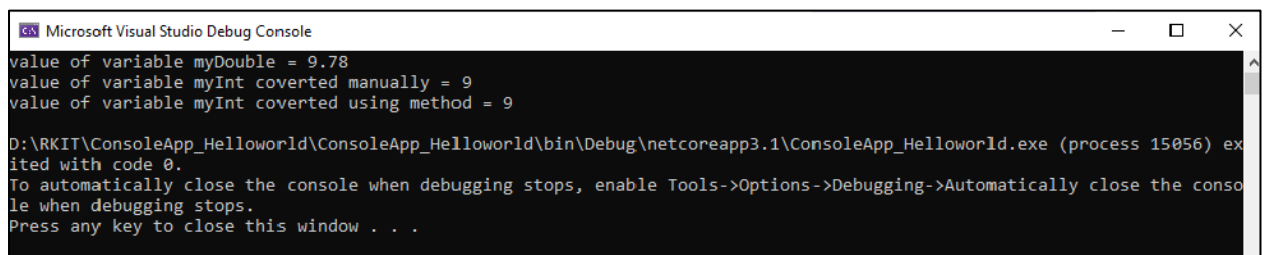
            Console.WriteLine("value of variable myDouble = " + myDouble);

            Console.WriteLine("value of variable myInt covered
manually = " + myInt);

            Console.WriteLine("value of variable myInt covered
using method = " + Convert.ToString(myInt));    // Explicit
casting using method: int to string

        }
    }
}
```

Output:



Microsoft Visual Studio Debug Console

```
value of variable myDouble = 9.78
value of variable myInt covered manually = 9
value of variable myInt covered using method = 9

D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 15056) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Boxing/Unboxing

Boxing

Boxing_Unboxing.cs

```
using System;

namespace ConsoleApp_Helloworld
{
    class Boxing_Unboxing
    {
        static void Main(string[] args)
        {
            int num = 2021;

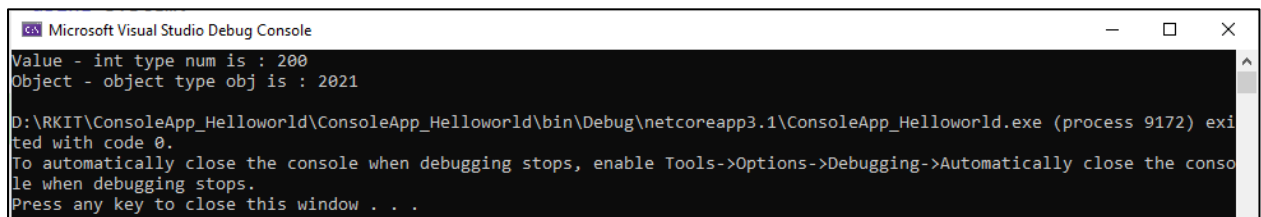
            // boxing
            object obj = num;

            // value of num to be change
            num = 200;

            System.Console.WriteLine("Value - int type num is : " + num);

            System.Console.WriteLine("Object - object type obj is : " + obj);
        }
    }
}
```

Output:



The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The console output is as follows:

```
Value - int type num is : 200
Object - object type obj is : 2021

D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 9172) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Unboxing

Boxing_Unboxing.cs

```
using System;

namespace ConsoleApp_Helloworld
{
    class Boxing_Unboxing
    {
        static void Main(string[] args)
        {
            int num = 20;

            // boxing
            object obj = num;

            //unboxing
            int i = (int)obj;

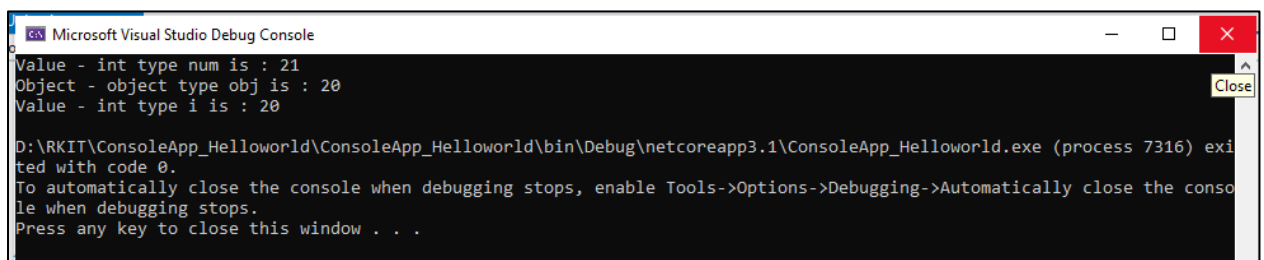
            //assigning new value to num
            num = 21;

            System.Console.WriteLine("Value - int type num is : " + num);

            System.Console.WriteLine("Object - object type obj is : " + obj);

            System.Console.WriteLine("Value - int type i is : " + i);
        }
    }
}
```

Output:



```
Microsoft Visual Studio Debug Console
Value - int type num is : 21
Object - object type obj is : 20
Value - int type i is : 20

D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 7316) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Understanding Decision making & statements

if-else

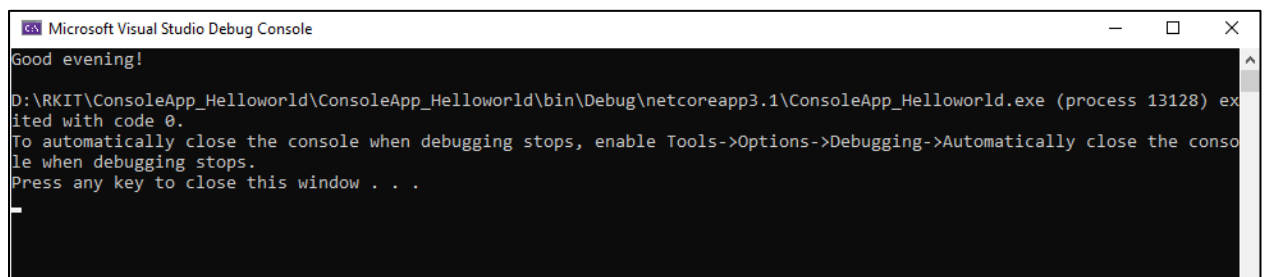
if-else statement

If_else.cs

```
using System;

namespace ConsoleApp_Helloworld
{
    class if_else
    {
        static void Main(string[] args)
        {
            int time = 20;
            if (time < 17)
            {
                Console.WriteLine("Have a good day!");
            }
            else
            {
                Console.WriteLine("Good evening!");
            }
        }
    }
}
```

Output:

A screenshot of the Microsoft Visual Studio Debug Console window. The window title is "Microsoft Visual Studio Debug Console". The output text is: "Good evening!". Below this, there is a message from the operating system: "D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 13128) exited with code 0." followed by instructions: "To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops." and "Press any key to close this window . . .". The console window has a dark background and a scrollbar on the right side.

```
Microsoft Visual Studio Debug Console
Good evening!
D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 13128) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

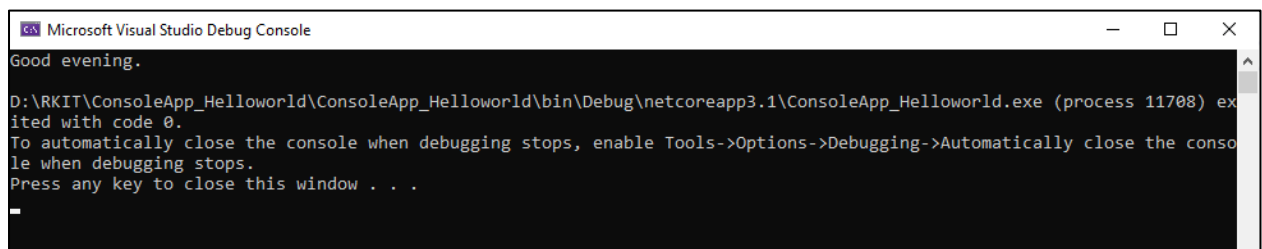
else-if statement

If_else.cs

```
using System;

namespace ConsoleApp_Helloworld
{
    class if_else
    {
        static void Main(string[] args)
        {
            int time = 22;
            if (time < 10)
            {
                Console.WriteLine("Good morning.");
            }
            else if (time < 20)
            {
                Console.WriteLine("Good day.");
            }
            else
            {
                Console.WriteLine("Good evening.");
            }
        }
    }
}
```

Output:



The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The console output is as follows:

```
Good evening.
D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 11708) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

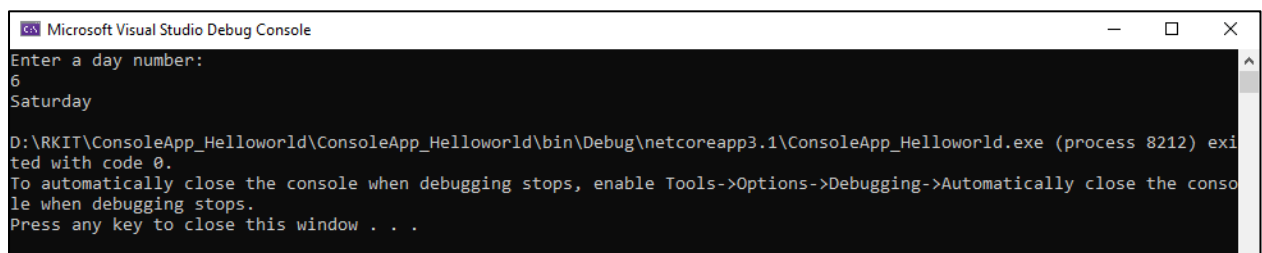

Switch

Switch_case.cs

```
using System;

namespace ConsoleApp_Helloworld
{
    class switch_case
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Enter a day number:");
            int day = Convert.ToInt32(Console.ReadLine());
            switch (day)
            {
                case 1:
                    Console.WriteLine("Monday");
                    break;
                case 2:
                    Console.WriteLine("Tuesday");
                    break;
                case 3:
                    Console.WriteLine("Wednesday");
                    break;
                case 4:
                    Console.WriteLine("Thursday");
                    break;
                case 5:
                    Console.WriteLine("Friday");
                    break;
                case 6:
                    Console.WriteLine("Saturday");
                    break;
                case 7:
                    Console.WriteLine("Sunday");
                    break;
                default:
                    Console.WriteLine("Invalid input!");
                    break;
            }
        }
    }
}
```

Output:



The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The console output is as follows:

```
Enter a day number:
6
Saturday

D:\RKIT\ConsoleApp_Helloworld\ConsoleApp_Helloworld\bin\Debug\netcoreapp3.1\ConsoleApp_Helloworld.exe (process 8212) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```