

# C#.NET SIMPLIFIED



**Mohammed Manzoor Ahmed**  
[www.manzoorhetrainer.com](http://www.manzoorhetrainer.com)

# CONTENTS

<b>ABOUT THE AUTHOR .....</b>	<b>4</b>
<b>CHAPTER 1: ARITHMETIC OPERATIONS .....</b>	<b>5</b>
<b>CHAPTER 2: TYPE CASTING .....</b>	<b>15</b>
<b>CHAPTER 3: CONTROL STRUCTURES .....</b>	<b>23</b>
<b>1. CONDITIONAL:.....</b>	<b>24</b>
IF AND IF-ELSE CONDITION:.....	24
ELSE-IF LADDER:.....	29
SWITCH CASE: .....	36
<b>2. ITERATIVE STATEMENTS OR LOOPS:.....</b>	<b>41</b>
FOR LOOP:.....	41
WHILE LOOP:.....	48
DO-WHILE LOOP: .....	53
<b>CHAPTER 4: ARRAYS .....</b>	<b>57</b>
<b>CHAPTE 5 : UNDERSTANDING THE CONCEPT OF FOREACH LOOP AND SYSTEM.ARRAY METHODS IN C#. .....</b>	<b>64</b>
<b>CHAPTER 6 : UNDERSTANDING THE CONCEPT OF STRUCTURES .....</b>	<b>75</b>
<b>CHAPTER 7 : CLASS .....</b>	<b>90</b>
<b>CHAPTER 8 : METHOD OVERLOADING OR STATIC POLYMORPHISM.....</b>	<b>99</b>
<b>CHAPTER 9 : THIS KEYWORD IN C#:.....</b>	<b>106</b>
<b>CHAPTER 10 : STATIC VARIABLE AND STATIC CONSTRUCTOR. ....</b>	<b>116</b>
<b>CHAPTER 11 : STATIC METHODS AND STATIC METHOD OVERLOADING .....</b>	<b>122</b>
<b>CHAPTER 12 : PROPERTIES IS C#.....</b>	<b>129</b>
<b>CHAPTER 13 : NAMESPACES IN C#.....</b>	<b>140</b>

<b>CHAPTER 14 : UNDERSTANDING THE CONCEPT OF INHERITANCE AND PROTECTED VARIABLES IN C# .....</b>	<b>148</b>
<b>CHAPTER 15 : CONSTRUCTOR CHAINING IN C#.....</b>	<b>155</b>
<b>CHAPTER 16 : METHOD OVERRIDING, VARIOUS TYPES OF INHERITANCE AND CONSTRUCTOR CHAINING IN C# .....</b>	<b>168</b>
<b>CHAPTER 17 : ABSTRACT METHOD AND ABSTRACT CLASS IN C# .....</b>	<b>184</b>
<b>CHAPTER 18 : REAL-TIME RUNTIME POLYMORPHISM IMPLEMENTATION IN C#.....</b>	<b>191</b>
<b>CHAPTER 19 : SEALED METHOD AND SEALED CLASS IN C# .....</b>	<b>207</b>
<b>CHAPTER 20 : INTERFACES AND MULTIPLE INHERITANCE IN C# .....</b>	<b>213</b>
<b>CHAPTER 21 : COLLECTION CLASSES AND ITS LIMITATION IN C# .....</b>	<b>222</b>
<b>CHAPTER 22 : GENERIC COLLECTION CLASSES IN C#.....</b>	<b>232</b>
<b>CHAPTER 23 : CONCEPT OF DELEGATES, UNICAST DELEGATES AND MULTICAST DELEGATES IN C# .....</b>	<b>239</b>
<b>CHAPTER 24 : DELEGATES, CALLBACK AND METHOD CONCEPT IN C# .....</b>	<b>247</b>

## ABOUT THE AUTHOR

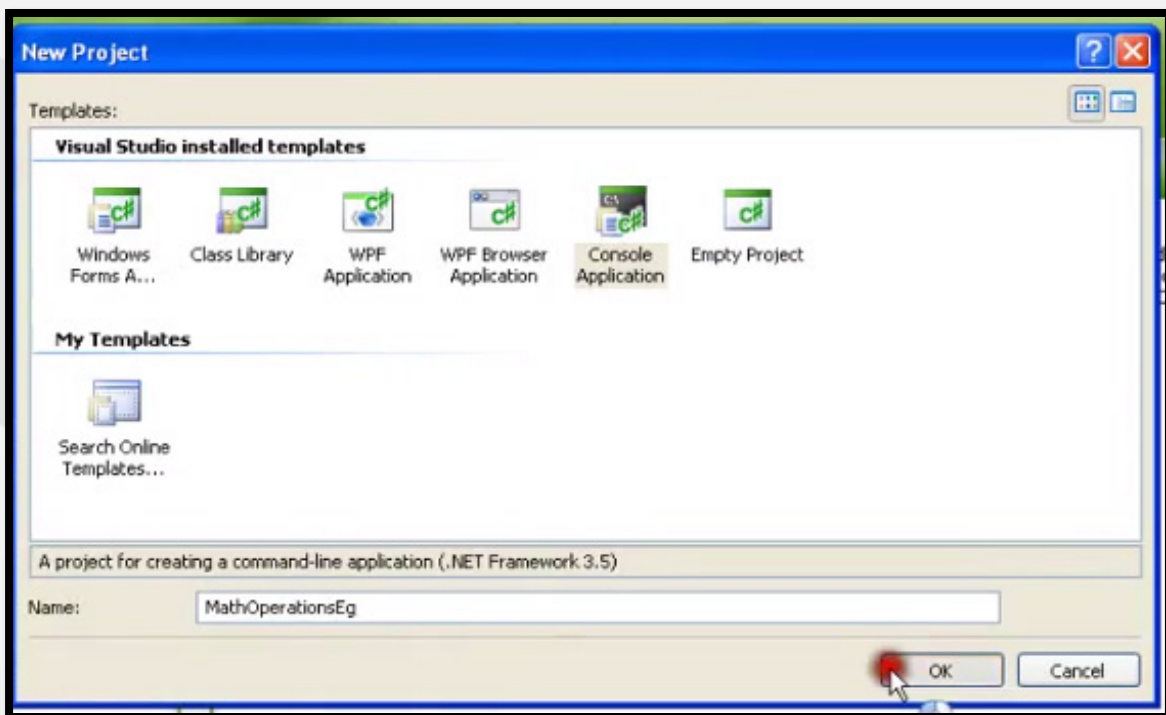
Manzoor is a Microsoft Certified Trainer who has been working on MS .Net technologies for more than a decade. Apart from development he is also passionate about delivering training on various MS .Net technologies and he has 10+ years of experience as a software development teacher. He writes articles for code-project as well. His YouTube channel has 1 million hits. He is the founder of ManzoorTheTrainer portal.

"I focus on simplifying, complex concepts..."

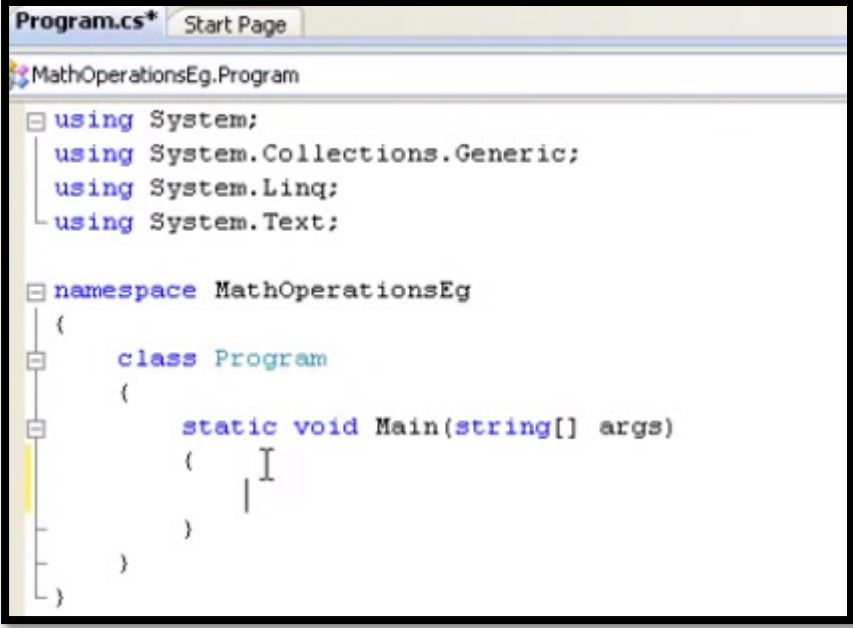
- ManzoorTheTrainer

## CHAPTER 1: ARITHMETIC OPERATIONS

- ◉ In this chapter we are going to see few arithmetic operations that we can perform using C#.
- ◉ Arithmetic operations like adding, subtracting, division, multiplication etc.
- ◉ Goto programs → Visual Studio.
- ◉ Select new project → Console application and name the project.
- ◉ Select Ok.



- ◉ Now it opens new window with Program.cs file name (i.e. Default page whenever we start console application).
- ◉ That program contain namespace as your file name (MathOperationsEg).
- ◉ In this page we need to start writing the program from main.



```
Program.cs* Start Page
MathOperationsEg.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace MathOperationsEg
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

- We have three variables (n1, n2, n3) of integers.
- Our intention is to store 67 in n1, 56 in n2 and add n1 and n2 values and store in n3.
- `int n1,n2,n3;`
- `n1=67;`
- `n2=56;`
- `n3=n1+n2;`
- We need to display the result as n3.
- To display the result we have method `Console.WriteLine(n3);`
- We need not to write n3 in quotation because we want to display the value of n3.

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

namespace MathOperationsEg
{
    class Program
    {
        static void Main(string [] args)
        {
            int n1, n2, n3;
            n1=67;
            n2=56;
            n3 = n1 + n2;
            Console.WriteLine(n3);
            Console.ReadLine();
        }
    }
}
```

- And ends with Console.ReadLine() method.
- To execute simply press F5.
- Result is 123 as shown below.

```
file:///C:/Documents and Settings/makvin/Local Settings/Application Data/Temporary Proje...
123
```

- ⦿ Now we have to display the result as sum of n1 and n2 is n3.
- ⦿ We want to display the result with meaningful message.
- ⦿ We'll just put this in quotation inside the Console.WriteLine () method.
- ⦿ Console.WriteLine("Sum of n1 and n2 is n3");
- ⦿ Press F5.

```
file:///C:/Documents and Settings/makvin/Local Settings/App
Sum of n1 and n2 is n3
_
```

- ⦿ Output is same as message inside the method.
- ⦿ Instead of this we want to display the values of n1, n2 and n3.
- ⦿ To display the values of n1, n2 and n3 simply we need to replace the n1, n2 and n3 with {0}, {1} and {2} as indexes in quotation.
- ⦿ Now put comma after quotation and give the variable name at 0<sup>th</sup> position (i.e. n1), n2 at 1<sup>st</sup> position and n3 at 3<sup>rd</sup> position.
- ⦿ Console.WriteLine("sum of {0} and {1} is {2}",n1,n2,n3);



```

static void Main(string [] args)
{
    int n1, n2, n3;

    n1=67;

    n2=56;

    n3 = n1 + n2;

    Console.WriteLine( "Sum of {0} and {1} is {2}",n1,n2,n3);

    Console.ReadLine();

}

```

- Press F5.

A screenshot of a Windows console window. The title bar reads 'file:///C:/Documents and Settings/makvin/Local Settings/Application Data/Temporary Proje...'. The console output shows 'Sum of 67 and 56 is 123' on the first line, followed by a blank line.

- There is another way to display the same result using '+' operator.
- It will not work as mathematical operator but it works as string concatenation.
- Console.WriteLine("sum of "+n1+" and "+n2+" is "+n3);

A screenshot of a Windows console window. The title bar reads 'file:///C:/Documents and Settings/makvin/Local Settings/Application Data/Temporary Proje...'. The console output shows 'Sum of 67 and 56 is 123' on the first line, and 'Sum of 67 and 56 is 123' on the second line.

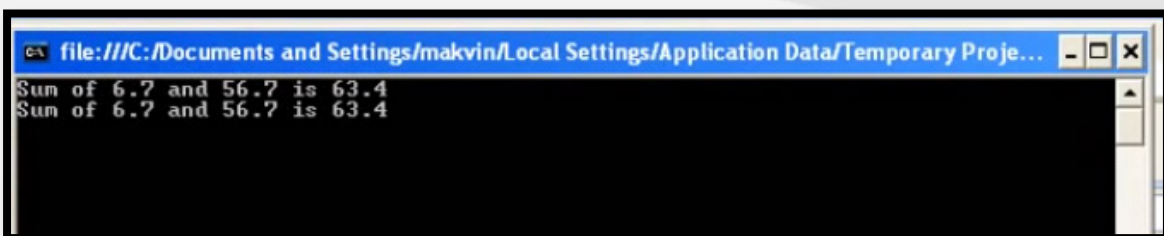
- Both of the methods display same result but using two different techniques.
- One is passing parameters kind of things like giving indexes for n number of variables, and another is using '+' operator for string concatenation.

- ⦿ Above program we get addition of two integer numbers if we give variable type as int.
- ⦿ If we want to give some decimals to perform addition then we need to change the variable type 'int to double'.
- ⦿ Double n1, n2, n3;

```
static void Main(string [] args)
{
    Double n1, n2, n3;
    n1=6.7;
    n2=56.7;
    n3 = n1 + n2;
    Console.WriteLine("Sum of {0} and {1} is {2}",n1,n2,n3);
    Console.WriteLine("Sum of " + n1 + " and " + n2 + " is " + n3);

    Console.ReadLine();
}
```

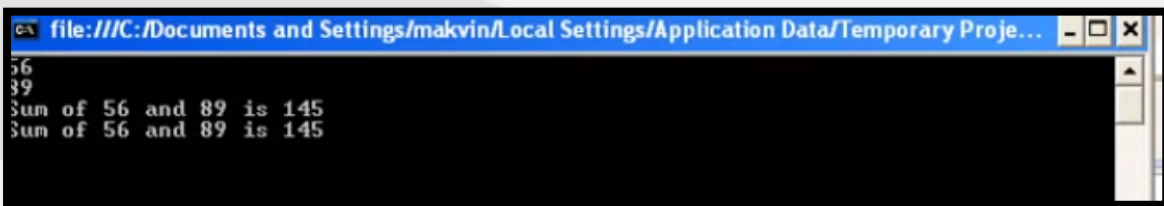
- ⦿ Press F5.



```
file:///C:/Documents and Settings/makvin/Local Settings/Application Data/Temporary Proje...
Sum of 6.7 and 56.7 is 63.4
Sum of 6.7 and 56.7 is 63.4
```

- ⦿ If we want to add different values we need not to go and edit the program again and again.
- ⦿ We can give the option to end user to enter the values from keyboard at runtime.

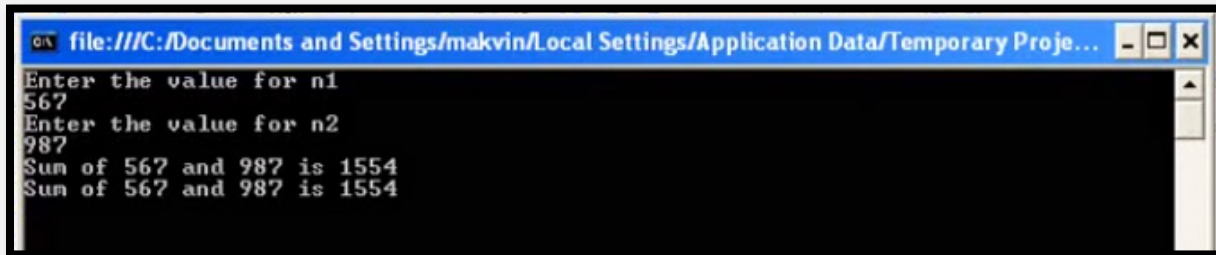
- For implementing this we can replace the values with `Console.ReadLine()` method means reading values from keyboard at runtime.
- But values read from keyboard at runtime is always in string format.
- And our variables (n1, n2) are in int or double type.
- So we need to convert this string type to integer type.
- We can achieve this by using `int.Parse(Console.ReadLine());` (means parse this string to Int).
- `n1= int.Parse(Console.ReadLine());`
- `n2= int.Parse(Console.ReadLine());`
- If we execute this it will be waiting for taking two numbers.
- Enter the two numbers.
- Press enter for result.



```
56
89
Sum of 56 and 89 is 145
Sum of 56 and 89 is 145
```

- For end users easy understandability we can give messages before entering the values from keyboard.
- First message as “Enter the value for n1”.
- And second message as “Enter the value for n2”.
- `Console.WriteLine(“Enter the value for n1”);`
- `n1= int.Parse(Console.ReadLine());`
- `Console.WriteLine(“Enter the value for n2”);`
- `n2= int.Parse(Console.ReadLine());`

- ◉ Remaining part is same.
- ◉ Press F5.



```
file:///C:/Documents and Settings/makvin/Local Settings/Application Data/Temporary Proje...
Enter the value for n1
567
Enter the value for n2
987
Sum of 567 and 987 is 1554
Sum of 567 and 987 is 1554
```

- ◉ If we have double values we need to change int type to double and instead of int.Parse we need to use double.Parse.
- ◉ Double n1,n2,n3;
- ◉ Console.WriteLine("Enter the value for n1");
- ◉ n1= double.Parse(Console.ReadLine());
- ◉ Console.WriteLine("Enter the value for n2");
- ◉ n2= double.Parse(Console.ReadLine());
- ◉ Complete code of addition of two integer numbers given below.

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

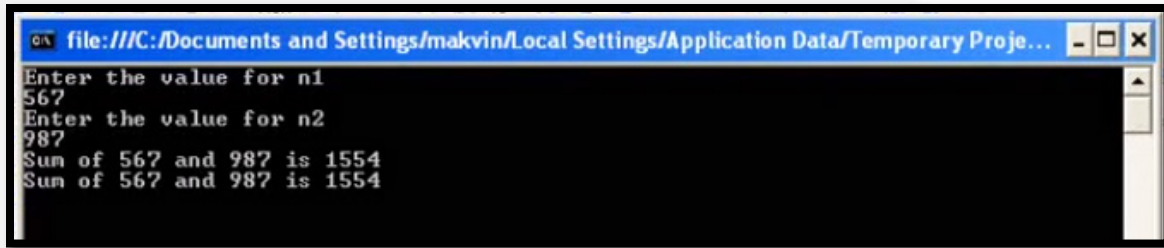
namespace MathOperationsEg
{
    class Program
    {
        static void Main(string [] args)
        {
            int n1, n2, n3;
            Console.WriteLine("Enter the value for n1");
            n1 =int.Parse(Console.ReadLine());

            Console.WriteLine("Enter the value for n2");
            n2 =int.Parse( Console.ReadLine());

            n3 = n1 + n2;
            Console.WriteLine("Sum of {0} and {1} is {2}",n1,n2,n3);
            Console.WriteLine("Sum of " + n1 + " and " + n2 + " is " + n3);

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

- Press F5.



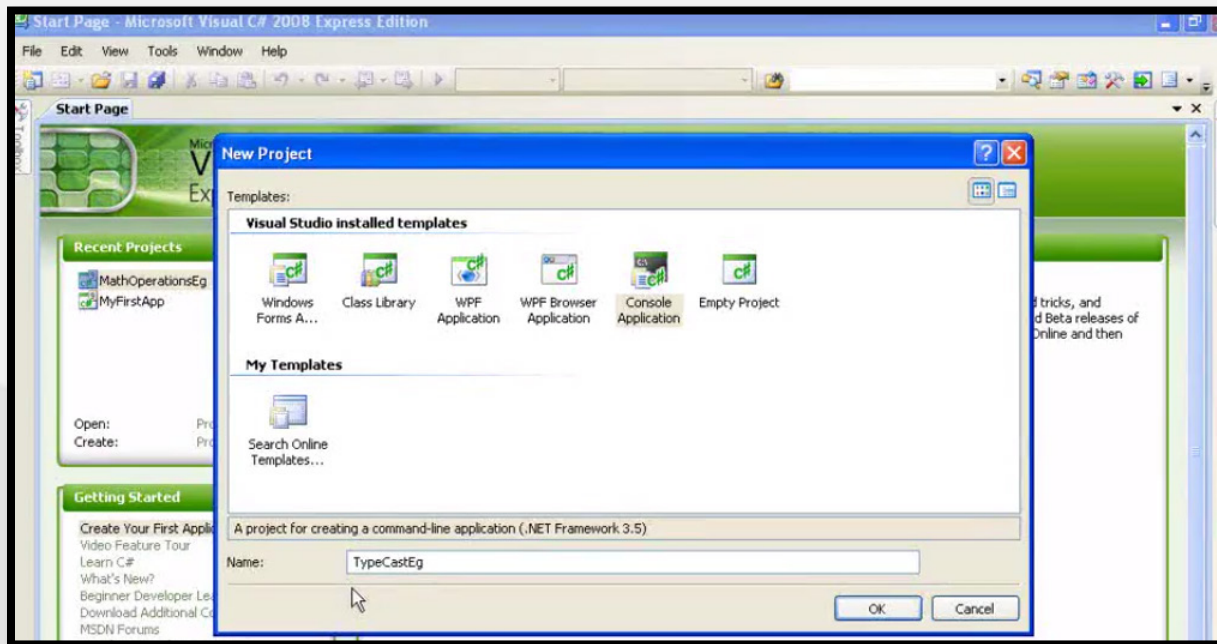
```
file:///C:/Documents and Settings/makvin/Local Settings/Application Data/Temporary Proje...
Enter the value for n1
567
Enter the value for n2
987
Sum of 567 and 987 is 1554
Sum of 567 and 987 is 1554
```

- ⦿ In this chapter we have seen addition of two integers and double numbers.
- ⦿ Display methods using various techniques.
- ⦿ Reading values from keyboard.
- ⦿ Int.Parse and double.Parse for conversion.

Thank you..!

## CHAPTER 2: TYPE CASTING

- ◉ In this chapter we'll see the concept of type casting.



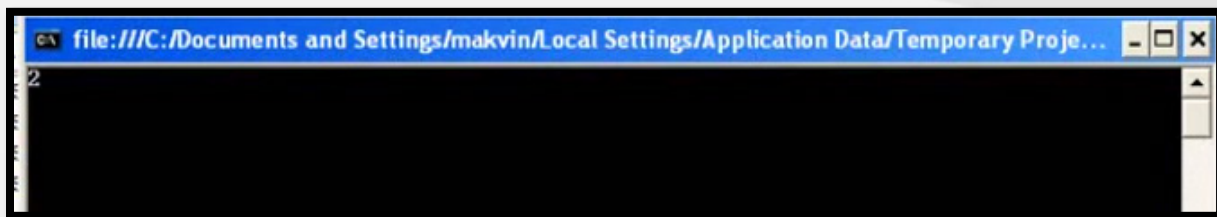
- ◉ Select new project → console application
- ◉ Name it as TypeCastEg click on ok.
- ◉ For example we have two integer variable a and b.
- ◉ `Int a=5;`
- ◉ `Int b=2;`
- ◉ If we say `console.writeline(a/b)`, could you guess the output?.
- ◉ Definitely we expect the output to be 2.5 as per our mathematical operations.
- ◉ Let us see what it says the result here.

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

namespace TypeCastEg
{
    class Program
    {
        static void Main(string [] args)
        {
            int a = 5;
            int b = 2;

            Console.WriteLine(a/b);
            Console.ReadLine();
        }
    }
}
```

- We'll just press F5.



- It says that result is 2.
- Because if both variables are integer our result also be an integer.
- If we observe this thing.



- ⦿  $5+2=7$ .
- ⦿ If we say  $5.0+2=7.0$ .
- ⦿ See that there is not much difference in mathematical concept 7 and 7.0 both are same.
- ⦿ Whereas in our computer world 7 and 7.0 has lot of difference in their data structure.
- ⦿ If we say  $5+2.0$  definitely our result would be 7.0.
- ⦿ If we say  $5.0+2.0$  our result would be again same as 7.0.
- ⦿ One thing we need to observe here is '5' is our first operand '2' is our second operand and '+' called as operator.
- ⦿ If our both the operands are integer our result is integer.
- ⦿ One of the two operands is double then our result is double.
- ⦿ If anyone operand is double our result will be double.
- ⦿ In the same way we are performing  $5/2$  the result will be 2.
- ⦿ Why because '5' is an integer '2' is an integer, so integer/integer will gives rise to integer?
- ⦿ We want the result to be double what is that we need to do, we need to make either 5 as double or 2 as double or both of them as double.
- ⦿ We'll make 5 as double.
- ⦿  $5.0/2$  this will give the result as 2.5.

## Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

