Input

In any programming language input means to feed some data into program. This can be given in the form of file or from command line. C# programming language provides a set of built-in methods to read given input and feed it to the program as per requirement.

System package/Class

C# Input, in the simplest method to get input from the user is by using the ReadLine() method of the Console class. However, Read() and ReadKey() are also available for getting input from the user. They are also included in Console class.

The Console class is used to get user input, and it is found in the *C# System* package. The Console class input of the primitive types like char and strings. It is the easiest way to read input in a C# program. In our example, we will use the WriteLine() method, which is used to read Strings:

 Syntax: variable =ReadLine() or variable = ReadKey()   or variable = Read()   etc.

Some Common Input Console class Methods in C# at Console environment:

Function	Purpose
Read()	This method reads the next character from the C# input stream – or command line – and returns the ASCII value of that character but variable should be integer
ReadLine( )	Reads a String value from the user

Example

```
using System;
namespace InputOutPutData
{
    class Program
    {
        static void Main(string[] args)
        {
            string name;
            int  ch;

            Console.Write("Enter any Name : ");
            name = Console.ReadLine();

            Console.Write("Enter any character : ");
            ch = Console.Read();

          Console.WriteLine("\n\nEnter your name is  " + name);
          Console.WriteLine("ASCII code of this character is  " + ch);
        }
    }
  }
```

Output:

In any programming language output means to display some data on screen, printer or in any file. C# programming language provides a set of built-in functions to output required data. We can use ***Console.Write()*** and ***Console.WriteLine()*** to display the data at console Screen. ***The*** main difference between WriteLine() and Write() is that the Write() method only prints the string provided to it, while the WriteLine() method prints the string and moves to the start of next line as well***.***

Some Common Output Method / Functions in C# of Console class:

Function	Purpose
Console.Write(String s)	Prints a string and cursor in same position of screen.
Console.WriteLine(String s)	Prints a string and cursor move on next line.

Type Casting

In C#, type casting is a method or process that converts a data type into another data type in both ways manually and automatically. The automatic conversion is done by the compiler and manual conversion performed by the programmer. In this section, we will discuss type casting and its types with proper examples.

1. Implicit Type Casting
2. Explicit Type Casting

Implicit Type Casting

In C# Language Converting a lower data type into a higher one is called implicit type casting. It is also known as widening conversion or casting down. It is done automatically. It is safe because there is no chance to lose data.
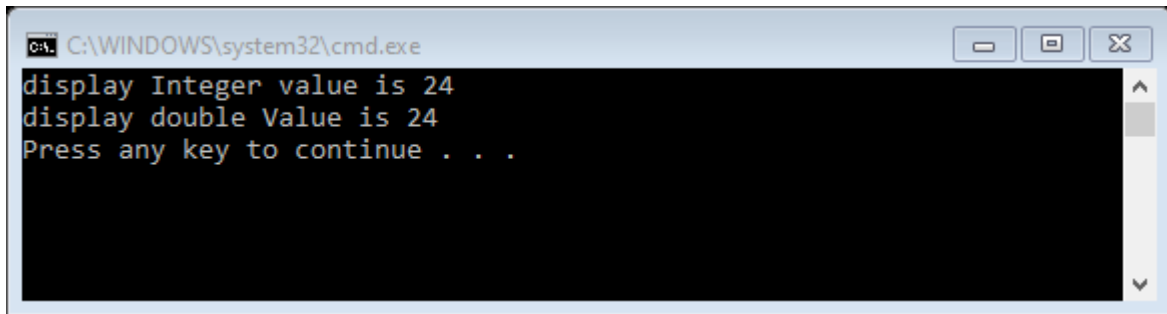
byte -> short -> char -> int -> long -> float -> double

Example

```csharp
using System;

namespace ImplicitTypeCasting
{
    class Program
    {
        static void Main(string[] args)
        {
          int numberInt = 24;
          double numberDouble = numberInt; // Automatic casting: int to double

           Console.WriteLine("display Integer value is " + numberInt);   // Outputs 24  4 bytes ocupied
           Console.WriteLine("display double Value is " + numberDouble); // Outputs 24 8 bytes ocupied

        }
    }
}
```

```
C:\WINDOWS\system32\cmd.exe
display Integer value is 24
display double Value is 24
Press any key to continue . . .
```

1.   Explicit Type Casting

In C# converting a higher data type into a lower one is called explicit type casting. It is also known as narrowing conversion or casting up. It is done manually by the programmer. If we do not perform casting then the compiler reports a compile-time error.

double -> float -> long -> nt -> short -> byte -> char

```csharp
using System;
namespace narrowingTypeCasting
{
    class Program
    {
        static void Main(string[] args)
        {
            double numberDouble = 1939.5678;
            int numberInt = (int)numberDouble; // Manual casting: double to int

            Console.WriteLine(numberDouble);   // Outputs 1939.5678
            Console.WriteLine(numberInt);      // Outputs 1939

        }
    }
}
```

Some cases we input data different numeric data type like integer, long, float and double etc. but input data in string or text form and after input we have convert data into required different numeric data types.

It is also possible to convert data types explicitly by using built-in methods, such as Convert.ToBoolean, Convert.ToDouble, Convert.ToString, Convert.ToInt32 (int) and Convert.ToInt64 (long): there for in this situation example are following.

```csharp
using System;

namespace TypeCasting
{
    class Program
    {
        static void Main(string[] args)
        {
            int roll;
            float per;
            string text;

            Console.Write(" Enter Student Roll Number …. :");
            text = Console.ReadLine();
            roll = Convert.ToInt32(text);

            Console.Write(" Enter Student Percentage ….. :");
            text = Console.ReadLine();
            per = Convert.ToSingle(text);

            Console.WriteLine("\n Student Roll Number is s "+roll);
            Console.WriteLine(" Student Percentage is   "+per);


        }
    }
}
```
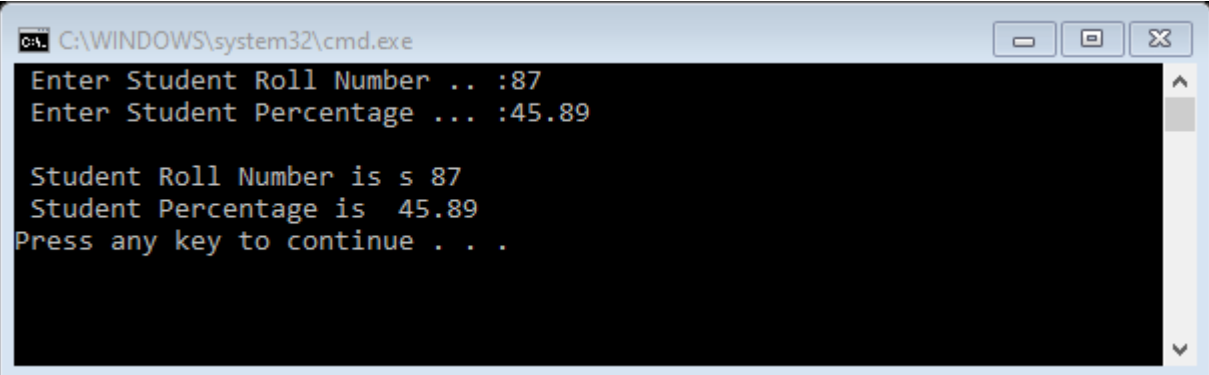
```
C:\WINDOWS\system32\cmd.exe                                    [_][□][X]
Enter Student Roll Number .. :87
Enter Student Percentage ... :45.89

Student Roll Number is s 87
Student Percentage is   45.89
Press any key to continue . . .
```

Exercise

Theory Question

Q1) what do mean by input data in programming.

Q2) what is difference between the *Console.Write*() and *Console.WriteLine*().

Q3) what is Type Casting and its types in C# languages.

Practical Question

Q1) Write a simple program of following using input and output methods/ functions:

Enter Student G.R Number:
Enter Student Name:
Enter Student Percentage:
Enter Student Class Section:
// Output below will display on the screen

Q2) Write a program to input radius and calculate area of circle (Formula: Area = PI x Radius^2).

Q3) Write name and purpose three Type casting methods of C# language with Example.

Objective MCQ's

1. Which of the following methods we can use to input any type of values?
    a) ReadKey()
    a) ReadLine()
    b) Read()
    c) Input()

2. Using System mean is a _____.
    a) Class name
    b) Import library
    c) Import file
    d) Name of package.

3. The automatic conversion is done by the compiler conversion.

a) Implicit type casting.
b) Simple type casting.
c) explicit Type Casting
d) Automatic Type Casting

4. The Converting a lower data type into a higher one is called.
   a) Implicit type casting.
   b) Simple type casting.
   c) Explicit Type Casting
   d) Automatic Type Casting

5. Converting a higher data type into a lower one is called narrowing type casting. It is also known as.
   a) Implicit conversion.
   b) Simplest conversion.
   c) Explicit Conversion.
   d) Automatic conversion.

6. Which of the following methods we can use to convert decimal point type of values?
   a) Convert.ToInt16()
   b) Convert.ToInt32()
   c) Convert.ToFloat()
   d) Convert.ToSingle()

7. Which of the following methods we can use to Convert 16 bit integer values?
   a) Convert.ToInt16()
   b) Convert.ToInt32()
   c) Convert.ToFloat()
   d) Convert.ToSingle()

Data Types
1. Which of the following is a valid variable name?
   a) TotalSalary ;
   b) Total Salary;

c) $TotalSalary;

d)   Total-Salary;

2.   Which is the correct syntax for declaring a variable and assigning it a string?

    a)   Name="Muhammad";

    b)   Name=Muhammad;

    c)   "Muhmmad"= Name;

    d)   Name = "Muahammad"

3. How many decimal places does an integer store

    a)   One decimal

    b)   Two decimal

    c)   Three decimal

    d)   Integer does not store decimal places.

4.   Variable name could be starting which symbol.

    a)   _ (underscore)

    b)   @

    c) $

    d)   %

5.   Combine the two or more string or other value by using concatenation___ symbol

    a) &

    b) *

    c) +

    d) . (dot)

6.   String constant value

    a)   Must be enclosed in double quotes

    b)   Must be enclosed in commas

    c)   Must be enclosed in round parentheses

    d)   Must be square brackets

7. A constant is case-sensitive by default. By convention, constant identifiers are should be____.

    a) Lower case

    b) Camel case

    c) Upper case

    d) Normal case

8. We use the _____ keyword to create constant.

    a)   define()

    b)   Include

    c)   final

    d)   const

9. A loosely typed programming language _____.

    a)   Does not required data typed of a variable to be declared.

    b)   Requires data types of variables to be declared

    c)   Does not have variable

    d)   Does not have different data types.