

# Flow Control

Conditional & Iteration Statements

# Conditional Statements

- If / else statements
- Switch / case statements
- Conditional operator: `a ? b : c`

# If / else

```
if (condition)  
    someStatement
```

```
else if (anotherCondition)  
    anotherStatement
```

```
else  
    yetAnotherStatement
```

# If / else

```
if (condition)
{
    someStatements
}
else
{
    otherStatements
}
```

# Nested If Statements

```
if (condition)
{
    if (anotherCondition)
        ...
    else
        ...
}
```

# Switch / case

```
switch(role)
{
    case Role.Admin:
        ...
        break;
    case Role.Moderator:
        ...
        break;
    default:
        ...
        break;
}
```

```
using System;
```

```
namespace Conditionals
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            int hour = 10;
```

```
            if (hour > 0 && hour < 12)
```

```
                Console.WriteLine("It's morning.");
```

```
            else if (hour >= 12 && hour < 18)
```

```
                Console.WriteLine("It's afternoon.");
```

```
            else
```

```
                Console.WriteLine("It's evening.");
```

```
        }
```

```
    }
```

```
}
```

```
using System;

namespace Conditionals
{
    class Program
    {
        static void Main(string[] args)
        {
            int hour = 10;

            if (hour > 0 && hour < 12)
            {
                Console.WriteLine("It's morning.");
            }
            else if (hour >= 12 && hour < 18)
            {
                Console.WriteLine("It's afternoon.");
            }
            else
            {
                Console.WriteLine("It's evening.");
            }
        }
    }
}
```



```
namespace Conditionals
{
    class Program
    {
        static void Main(string[] args)
        {
            bool isGoldCustomer = true;

            float price;
            if (isGoldCustomer)
                price = 19.95f;
            else
                price = 29.95f;
        }
    }
}
```

```
using System;
```

```
namespace Conditionals
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            bool isGoldCustomer = true;
```

```
            //float price;
```

```
            //if (isGoldCustomer)
```

```
            //    price = 19.95f;
```

```
            //else
```

```
            //    price = 29.95f;
```

```
            float price = (isGoldCustomer) ? 19.95f : 29.95f;
```

```
            Console.WriteLine(price);
```

```
        }
```

```
    }
```

```
}
```

```
namespace Conditionals
{
    public enum Season
    {
        Spring,
        Summer,
        Autumn,
        Winter
    }

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

```
using System;
```

```
namespace Conditionals
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            var season = Season.Autumn;
```

```
            switch (season)
```

```
            {
```

```
                case Season.Autumn:
```

```
                    Console.WriteLine("It's autumn and a beautiful season.");  
                    break;
```

```
                case Season.Summer:
```

```
                    Console.WriteLine("It's perfect to go to beach.");  
                    break;
```

```
                default:
```

```
                    Console.WriteLine("I don't understand that season!");  
                    break;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
using System;
```

```
namespace Conditionals
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            var season = Season.Autumn;
```

```
            switch (season)
```

```
            {
```

```
                case Season.Autumn:
```

```
                    Console.WriteLine("We've got promotion");
```

```
                    break;
```

```
                case Season.Summer:
```

```
                    Console.WriteLine("We've got promotion");
```

```
                    break;
```

```
                default:
```

```
                    Console.WriteLine("I don't understand that season!");
```

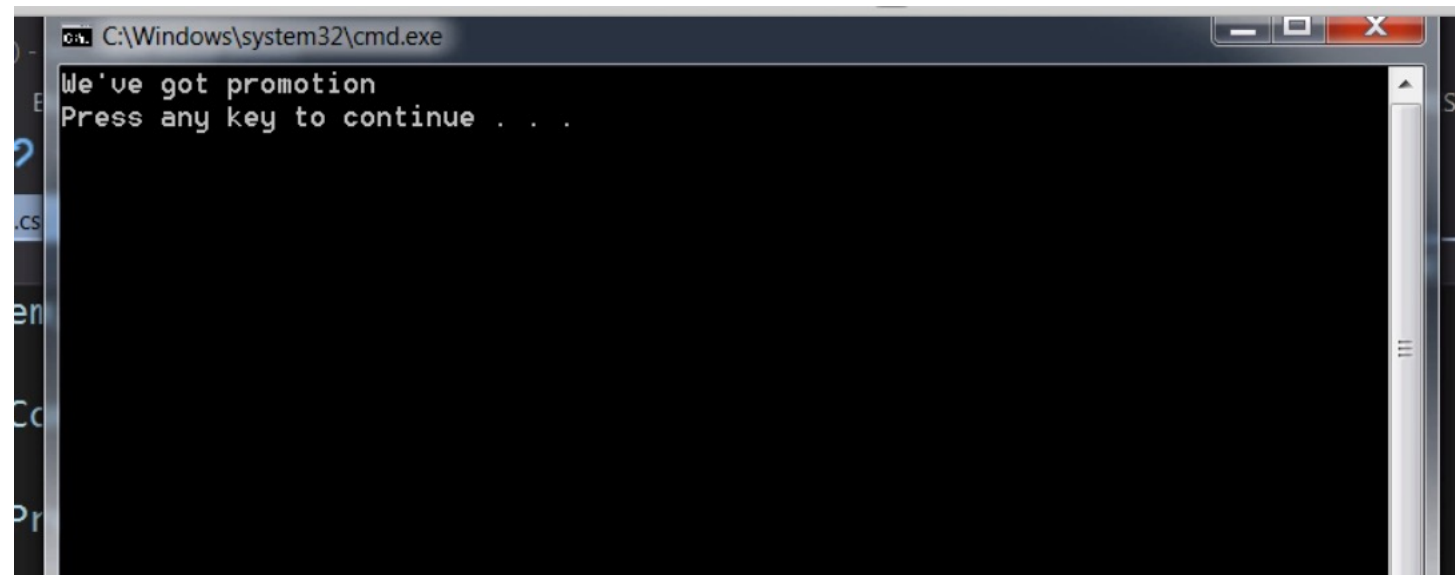
```
                    break;
```

```
            }
```

```
        }
```

```
    }
```





# Excercise

**Note:** for all these exercises, ignore input validation unless otherwise directed. Assume the user enters a value in the format that the program expects. For example, if the program expects the user to enter a number, don't worry about validating if the input is a number or not. When testing your program, simply enter a number.

- 1- Write a program and ask the user to enter a number. The number should be between 1 to 10. If the user enters a valid number, display "Valid" on the console. Otherwise, display "Invalid". (This logic is used a lot in applications where values entered into input boxes need to be validated.)
- 2- Write a program which takes two numbers from the console and displays the maximum of the two.
- 3- Write a program and ask the user to enter the width and height of an image. Then tell if the image is landscape or portrait.
- 4- Your job is to write a program for a speed camera. For simplicity, ignore the details such as camera, sensors, etc and focus purely on the logic. Write a program that asks the user to enter the speed limit. Once set, the program asks for the speed of a car. If the user enters a value less than the speed limit, program should display Ok on the console. If the value is above the speed limit, the program should calculate the number of demerit points. For every 5km/hr above the speed limit, 1 demerit points should be incurred and displayed on the console. If the number of demerit points is above 12, the program should display License Suspended.