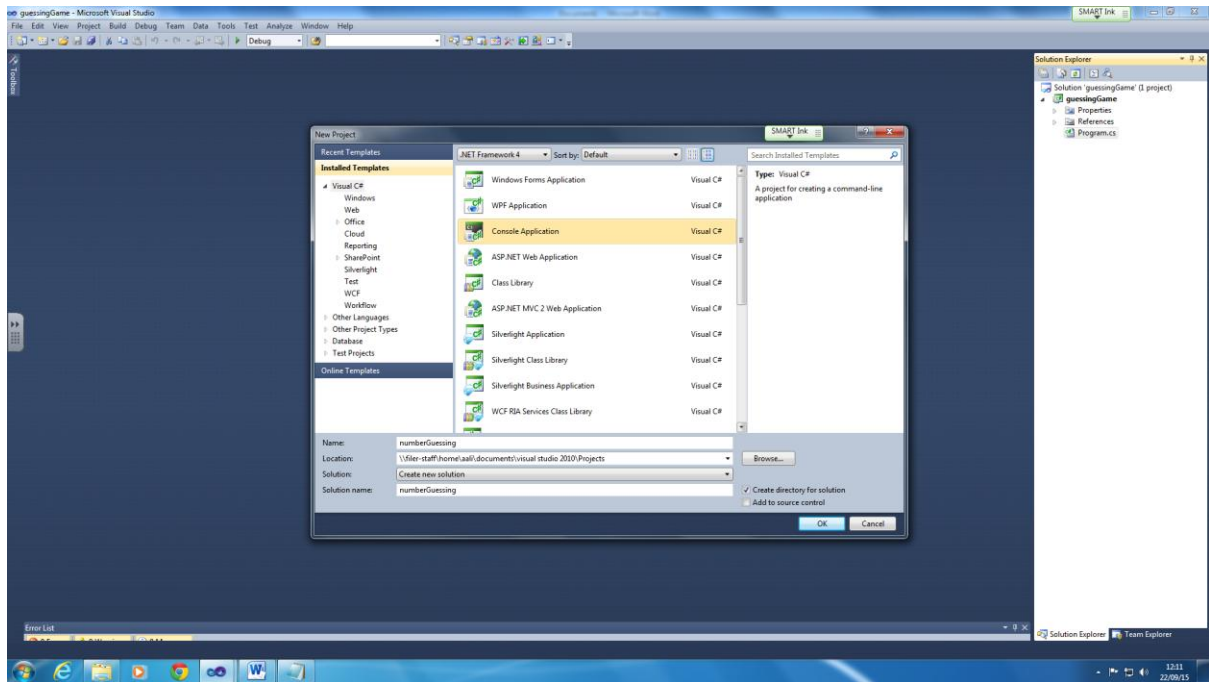


Number guessing game

Open a new project in visual studio

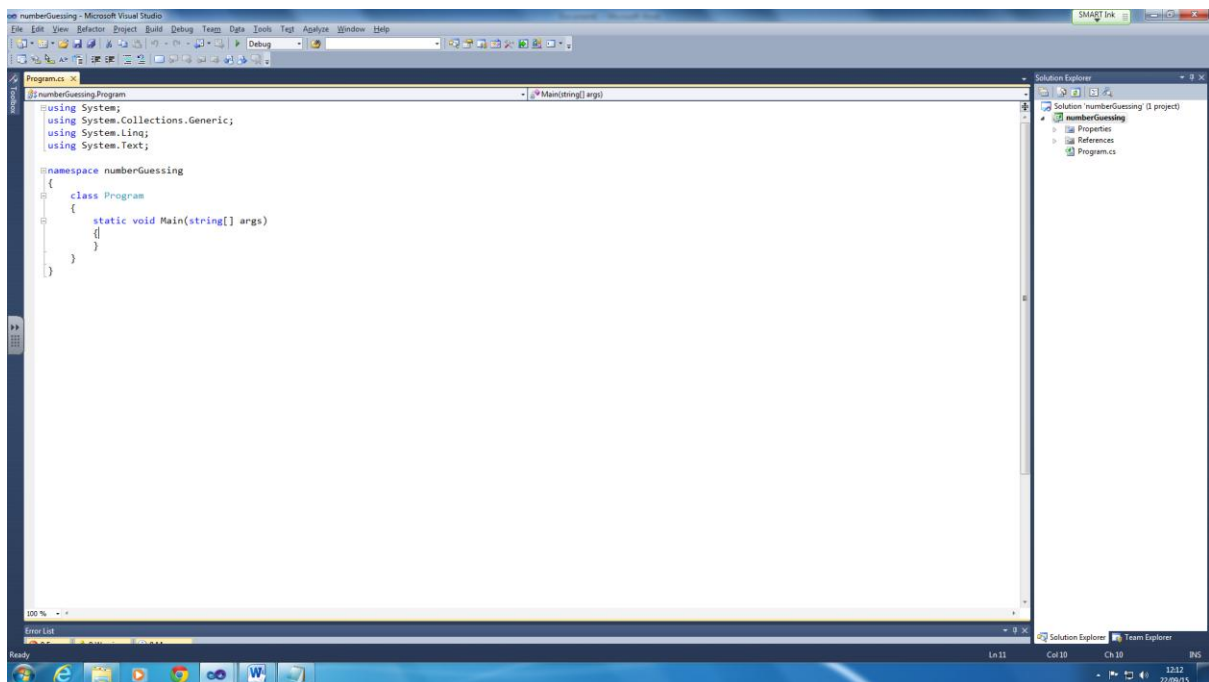
Choose console application

Call it numberGuessing

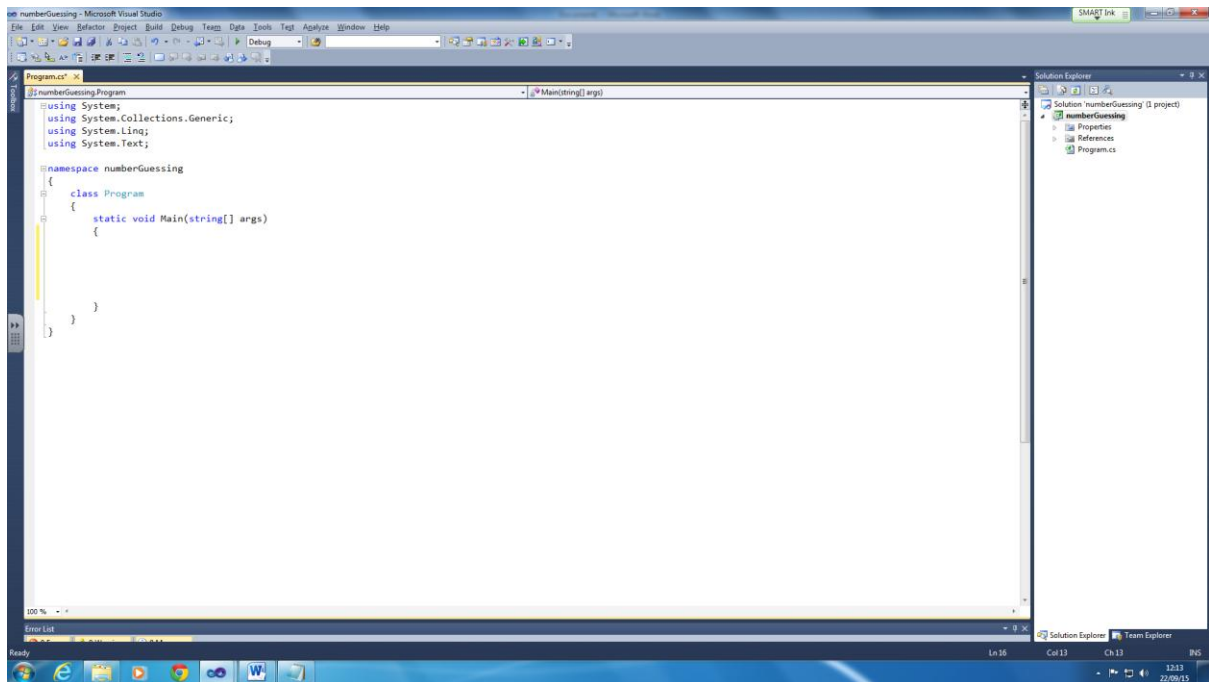


Click OK

All the programming we will be doing will go inside the Main function.



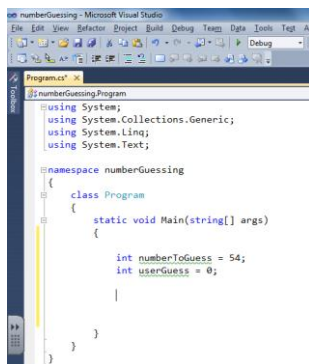
Make some space between the two curly brackets#



We need two different variables in for this program

`int numberToGuess = 54;` ← this is the number which the player will have to guess. You change this number to a number you want later on.

`int userGuess = 0;` ← what the user will guess will be stored here for comparison with `numberToGuess`.



Now we will enter a loop, specifically a while loop.

While(condition here)

{

While the loop is running the code will run from here

}

A while loop will run the loop until it finds the result it is looking for. This is a good loop to run for this program. It will compare the use guess against the actual number to figure out whether the user have guessed the right number or not.

```

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

namespace numberGuessing
{
    class Program
    {
        static void Main(string[] args)
        {
            int numberToGuess = 54;
            int userGuess = 0;

            while ()
            {
            }
        }
    }
}

```

Lets start adding things to our loop

First the condition for this loop

While (**userGuess != numberToGuess**)

```

{
}

```

In programming here are some conditional parameters

==	Equals to
!=	Not equals to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

We are using not equals to because the loop should only run if the user guessed it wrong or else they win the game thus end of the program. Neat right

```

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

namespace numberGuessing
{
    class Program
    {
        static void Main(string[] args)
        {
            int numberToGuess = 54;
            int userGuess = 0;

            while (userGuess != numberToGuess)
            {
            }
        }
    }
}

```

Now that's done we need several IF statements to ensure we are giving out the right message to the user and the user decisions are being appropriately processed.

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

namespace numberGuessing
{
    class Program
    {
        static void Main(string[] args)
        {
            int numberToGuess = 54;
            int userGuess = 0;

            while (userGuess != numberToGuess)
            {
                Console.WriteLine("Enter your guess: ");
            }
        }
    }
}
```

Add the console.write to tell the user to enter a number.

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

namespace numberGuessing
{
    class Program
    {
        static void Main(string[] args)
        {
            int numberToGuess = 54;
            int userGuess = 0;

            while (userGuess != numberToGuess)
            {
                Console.WriteLine("Enter your guess: ");
                int.TryParse(Console.ReadLine(), out userGuess);
            }
        }
    }
}
```

Now add **int.TryParse(Console.ReadLine(), out userGuess);**

What is this line doing. **Int.tryParse** we looked at before which is capturing the data inside that user input and converting it to a integer.

Console.ReadLine() is allowing the user data to be inputter.

out userGuess is storing the user input inside the userGuess variable.

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

namespace numberGuessing
{
    class Program
    {
        static void Main(string[] args)
        {
            int numberToGuess = 54;
            int userGuess = 0;

            while (userGuess != numberToGuess)
            {
                Console.Write("Enter your guess: ");
                int.TryParse(Console.ReadLine(), out userGuess);

                if ()
                {
                    |
                }
            }
        }
    }
}
```

Now lets add our first if statement

Once again we will needing a condition inside the if statement to work.

```
namespace numberGuessing
{
    class Program
    {
        static void Main(string[] args)
        {
            int numberToGuess = 54;
            int userGuess = 0;

            while (userGuess != numberToGuess)
            {
                Console.Write("Enter your guess: ");
                int.TryParse(Console.ReadLine(), out userGuess);

                if (userGuess > numberToGuess)
                {
                }
            }
        }
    }
}
```

userGuess > numberToGuess

This line is showing the condition IF user guess is greater than number to guess.



```
namespace numberGuessing
{
    class Program
    {
        static void Main(string[] args)
        {
            int numberToGuess = 54;
            int userGuess = 0;
            while (userGuess != numberToGuess)
            {
                Console.Write("Enter your guess: ");
                int.TryParse(Console.ReadLine(), out userGuess);

                if (userGuess > numberToGuess)
                {
                    Console.WriteLine("{0} is too high!", userGuess);
                }
            }
        }
    }
}
```

Add **Console.WriteLine("{0} is too high!", userGuess);**

If the number entered is higher than the original number than it will say your number is too high.

Then the user will be allowed to guess again.

Now we need an **else if** in the program for a second decision.

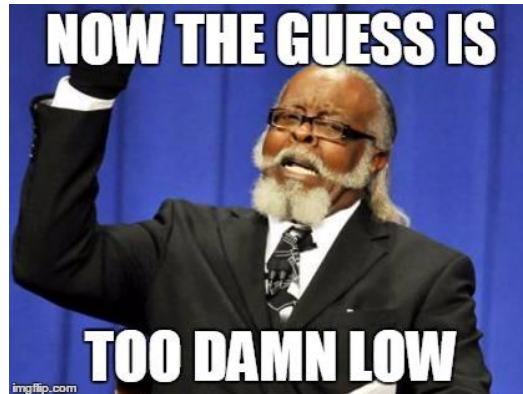
```
namespace numberGuessing
{
    class Program
    {
        static void Main(string[] args)
        {
            int numberToGuess = 54;
            int userGuess = 0;
            while (userGuess != numberToGuess)
            {
                Console.Write("Enter your guess: ");
                int.TryParse(Console.ReadLine(), out userGuess);

                if (userGuess > numberToGuess)
                {
                    Console.WriteLine("{0} is too high!", userGuess);
                }
                else if (userGuess < numberToGuess)
                {
                    Console.WriteLine("{0} is too low!", userGuess);
                }
            }
        }
    }
}
```

Add

```
elseif (userGuess < numberToGuess)
{
    Console.WriteLine("{0} is too low!", userGuess);
}
```

Now the lines above is doing exactly as the IF statement before but it's going the other direction. It's checking whether the user input is lower than the actual number. If it is then it will say your guess is too low.



Lastly we need a else statement to make sure the if the user guesses the right number then they win

```
static void Main(string[] args)
{
    int numberToGuess = 54;
    int userGuess = 0;
    while (userGuess != numberToGuess)
    {
        Console.Write("Enter your guess: ");
        int.TryParse(Console.ReadLine(), out userGuess);

        if (userGuess > numberToGuess)
        {
            Console.WriteLine("{0} is too high!", userGuess);
        }
        else if (userGuess < numberToGuess)
        {
            Console.WriteLine("{0} is too low!", userGuess);
        }
        else
        {
            Console.WriteLine("{0} is right! Congratulations.", userGuess);
        }
    }
}
```

Console.WriteLine("{0} is right! Congratulations.", userGuess);

The line above will then make sure the right message is shown on the screen.

Now test the program and double check the code.

With that being done.

In the last else where it says congratulations get the program to show this line below

```
CMD.EXE was started with the above path as the current directory.  
UNC paths are not supported.  Defaulting to Windows directory.  
Enter your guess: 54  
54 is right! Congratulations.  
Program created by YOUR NAME HERE  
Press any key to continue . . . _
```

LET THE PROGRAM SHOW YOU CREATED IN THE END.