

ref

Contents

- 1) What does “ref” mean
- 2) Situations where you want “ref”
- 3) Situations where you do not want “ref”
- 4) Example

Contents

- 1) What does “ref” mean
- 2) Situations where you want “ref”
- 3) Situations where you do not want “ref”
- 4) Example

1 What does “ref” mean

1 What does “ref” mean

- “ref” means reference

1 What does “ref” mean

- “ref” means reference
- it's the address of an object

1.1 Why do you need addresses

1.1 Why do you need addresses

- Generally, everything is copied when you pass it into a function

1.1 Why do you need addresses

- Generally, everything is copied when you pass it into a function
- You can choose...

1.1 Why do you need addresses

- Generally, everything is copied when you pass it into a function
- You can choose...
 - Do you want to copy a variable of size 1 GB

1.1 Why do you need addresses

- Generally, everything is copied when you pass it into a function
- You can choose...
 - Do you want to copy a variable of size 1 GB
 - Or do you want to copy its address, which is only 4 Bytes and tells where the variable is in memory

Contents

- 1) What does “ref” mean
- 2) Situations where you want “ref”
- 3) Situations where you do not want “ref”
- 4) Example

2 Situations where you want “ref”

2 Situations where you want “ref”

- When you want to work on the original object

2 Situations where you want “ref”

- When you want to work on the original object
- **Changes to an object which has been passed with “ref”, will also be changed outside the function**

Contents

- 1) What does “ref” mean
- 2) Situations where you want “ref”
- 3) Situations where you do not want “ref”
- 4) Example

3 Situations where you do not want “ref”

3 Situations where you do not want “ref”

- If you don't want your object to be changed outside your function

3 Situations where you do not want “ref”

- If you don't want your object to be changed outside your function
- You create a copy of your object and the function can do whatever to the object to do calculations e.g.

3 Situations where you do not want “ref”

- If you don't want your object to be changed outside your function
- You create a copy of your object and the function can do whatever to the object to do calculations e.g.
- **The original object will not be changed**

Contents

- 1) What does “ref” mean
- 2) Situations where you want “ref”
- 3) Situations where you do not want “ref”
- 4) Example

4 Examples

4 Examples

```
1  static void Main(string[] args)
2  {
3      int number = 12;
4      Console.WriteLine("number = " + number);
5      increaseByOne(number);
6      Console.WriteLine("number = " + number);
7  }
8
9  public static void increaseByOne(int number)
10 {
11     number = number + 1;
12 }
```

4 Examples

```
1  static void Main(string[] args)
2  {
3      int number = 12;
4      Console.WriteLine("number = " + number);
5      increaseByOne(number);
6      Console.WriteLine("number = " + number);
7  }
8
9  public static void increaseByOne(int number)
10 {
11     number = number + 1;
12 }
```

Output:

number = 12

number = 12

4 Examples

```
1  static void Main(string[] args)
2  {
3      int number = 12;
4      Console.WriteLine("number = " + number);
5      increaseByOne(ref number);
6      Console.WriteLine("number = " + number);
7  }
8
9  public static void increaseByOne(ref int number)
10 {
11     number = number + 1;
12 }
```

4 Examples

```
1  static void Main(string[] args)
2  {
3      int number = 12;
4      Console.WriteLine("number = " + number);
5      increaseByOne(ref number);
6      Console.WriteLine("number = " + number);
7  }
8
9  public static void increaseByOne(ref int number)
10 {
11     number = number + 1;
12 }
```

Output:

number = 12

number = 13