# 99 Bottles of OOP

by Sandy Metz and Katrina Owen

# The 99 Bottles Song

*The 99 Bottles Song*

*99 bottles of milk on the wall, 99 bottles of milk.*
*Take one down and pass it around, 98 bottles of milk on the wall.*

*98 bottles of milk on the wall, 98 bottles of milk.*
*Take one down and pass it around, 97 bottles of milk on the wall.*

*(…  … … … … …Keep repeating upto 3 bottles… … … … … … )*

*2 bottles of milk on the wall, 2 bottles of milk.*
*Take one down and pass it around, 1 bottle of milk on the wall.*

*1 bottle of milk on the wall, 99 bottles of milk.*
*Take it down and pass it around, no more bottles of milk on the wall.*

*No more bottles of milk on the wall, no more bottles of milk.*
*Go to the store and buy some more, 99 bottles of milk on the wall.*

```
public string Verse(int n)
{
    var verse = "";
    if (n == 0)
        verse += "No more";
    else
        verse += n;

    verse += " bottle";
    if (n != 1)
        verse += "s";

    verse += " of milk on the wall, ";

    if (n == 0)
        verse += "no more";
    else
        verse += n;

    verse += " bottle";
    if (n != 1)
        verse += "s";
    verse += " of milk.\n";
```

```
if (n > 0)
{
    verse += "Take ";
    if (n > 1)
        verse+= "one ";
    else
        verse+= "it ";

    verse+= "down and pass it around, ";
}
else
{
    verse+= "Go to the store and buy some more, ";
}
```

```
if (n-1 < 0)
{
    verse+= 99;
}
else
{
    if (n-1 == 0)
    {
        verse+= "no more";
    }
    else
    {
        verse+= (n-1);
    }
}

verse += " bottle";
if (n - 1 != 1)
    verse += "s";

verse+= " of milk on the wall.";

return verse;
}
```

## Problems

- Parameter n is poorly named.
- String verse could be a StringBuilder/StringBuffer
- Too many nested ifs
- The method is too long
- Same strings are repeated several times
- Same conditions are repeated several times

# Facts about software

1. Change request will be there, no matter what the software is.
    a. For client project, clients will be requesting change
    b. For public facing software
        i. users will be requesting new features
        ii. The company will analyze use cases and figure out required changes
    c. For your pet-project, you will be the one to make changes

    Implications of this is that **code should be easy to change**.

2. Code is read far more times than it is written. And in many cases, code is read by the authors themselves.

    Implication of this is that **code should be readable** (to others, of course, even to you). Even if writing readable code is hard, we should work hard to make sure reading is easy.

## Clean Code

# Clean Code

- paining analogy in Recognizing vs Writing Clean Code

**What is Clean Code**

Best clean is where nothing exists, but that clean is undesirable.

Attributes of a clean code, by priority -

- Easy to read
- Easy to change

**Refactoring**

Refactoring is the process of improving a code, without changing its original behavior.