

## Lab 11

Annie Hayes

Due: Thursday, April 22, 2021, by the end of your lab section.

You must submit lab through Canvas.

### Instructions:

Listed below is a series of short videos posted by the School of Computing's very own Dr. Jacob Sorber. Dr. Sorber is a great instructor and very knowledgeable in the C programming language. If you have the opportunity to take him for Operating Systems you should do so.

You are required to watch Dr. Sorber's "Code Smells" videos. There are 5 videos in this series. A link to each video is listed below along with a set of questions that you must answer for each video.

Your answers MUST be in **RED**. If your answers are not in red points will be deducted.

These videos are short but very informative.

### Code Smells

#### Segment 1 – Smelly code and Magic Numbers.

[https://www.youtube.com/watch?v=p8RC\\_i9t0MU](https://www.youtube.com/watch?v=p8RC_i9t0MU)

According to Dr. Sorber, what is a magic number?

According to Dr. Sorber, magic numbers are numbers that are unique in a certain way. These numbers have multiple occurrences or a meaning that is too complex to comprehend to not easily replace with a named constant.

Can you think of an example of a magic number.

```
//this is an example of a magic number (26)
//this function finds the average scores for 26 students
int main() {
    int n = 1;
    int scoresSum = 0;
    int avg = 0;
    if (n <= 26) {
        int temp = 0;
        printf("input score for student %d \n", n);
        scanf("%d", temp);
        scoresSum = scoresSum + temp; n++;
    }
    avg = scoresSum / 26;
}
```

What is the problem with Magic Numbers. He gives 2 problems what are they.

Magic Numbers don't stick numerical constants throughout the code unless it's necessary/obvious to do. The other problem is that it is easier to alter the value of the number because it is not duplicated hence indicating magic numbers hide intentions.

Dr. Sorber gave two reasons when he believes it is acceptable to use a magic number. Name one.

As Dr. Sorber explained, "When testing for even numbers, the use of 2 is probably ok because it's pretty obvious mathematically what this is doing and you're not likely to change it."

Dr. Sorber indicates using magic numbers is not always bad and discussed things you should consider when making your decision, with respect to the use of magic numbers. What where they?

The point is to not eliminate all constants from your code but to look at your code and find it easier to maintain and read.

## Code Smells

### Segment 2 – Duplicate Code

[https://www.youtube.com/watch?v=ck\\_RfVOYgjQ](https://www.youtube.com/watch?v=ck_RfVOYgjQ)

Why is duplicate code not your friend?

Because it makes your code longer than it needs to be. This can lead to more opportunities for bugs and makes your code harder to maintain. It also makes your code harder to change. This repeated logic requires a change in all of the places where you copy and pasted code.

In the videos, Dr. Sorber introduces the D.R.Y. principle. What does D.R.Y. stand for and what is his solution to this problem.?

D.R.Y. stands for do not repeat yourself. His solution is to dry up your code. If he tightens things up, moves things into a function, and then allows those functions to be called in each of the critical points, making things easier to follow.

Dr. Sorber states, the D.R.Y. principle makes your code shorter easier to read, understand, and debug.

He also states, the D.R.Y. principle makes your called, easier to follow and read and more likely to get the code right.

## Code Smells

### Segment 3 Bad Names

<https://www.youtube.com/watch?v=zx7euEEZ0H4>

Dr. Sorber mentioned several problems that can arise if you do not practice good naming conventions in your code. What were some of the issues that may arise for poor naming convention.

It would make it harder to read or harder to follow.

He listed some naming conventions that beginner students use that may or may not always be acceptable. Name at least 5 he listed and give an example of each. Do not just copy his example, think about these and give your own example.

Dr. Sorber listed examples such as foo, h, c, snake\_case, and CamelCase.

foo:

//this is an example of a bad naming convention (variable foo)

//this function check arguments

```
void check(int foo)
{
    //checks to validate that command line argument is functional
    if ( foo < 2)
    {
        printf("incorrect number of command line arguments!");
        exit(-1);
    }
}
```

h:

//this is an example of a bad naming convention (variable h)

//this function adds

```
int h(int first, int second)
{
    int h = first + second;
    return h;
}
```

c:

//this is an example of a bad naming convention (variable c)

//this program prints "Hello World"

```
void c()
{
    printf("Hello World\n");
}
```

snake\_case:

//this is an example of a good naming convention (variable check\_args)  
 //this function check arguments

```
void check_args(int check_num)
{
    //checks to validate that command line argument is functional
    if ( check_num < 2)
    {
        printf("incorrect number of command line arguments!");
        exit(-1);
    }
}
```

CamelCase:

//this is an example of a good naming convention (variable h)  
 //this function adds

```
int addVars(int firstNum, int secondNum)
{
    int sumOfNumbers = firstNum + secondNum;
    return sumOfNumbers;
}
```

When is it appropriate **not** be concerned with your naming convention and when **should** you be concerned with the naming convention?

You should not be concerned when you have a name that reflects what it does. You should be concerned when you are trying to make the function do too much.

When choosing names for your code Dr. Sorber listed 3 things you need to think about. List the three in their order of importance and give a short description of each.

Making names understandable (easy to understand), consistent (keeps it befitting) and short (something that isn't too long).

What are some of the examples of multiple word naming convention he mentioned?

Some examples are: `this_is_snake_case`, `CamelCase`, `SCREAMING_SNAKE_CASE`.

List the one that most reflects your style of naming convention.

`CamelCase` most reflects my style of naming convention.

Code Smells

## Segment 4 Comments

<https://www.youtube.com/watch?v=LLqDNjr0kPo>

This video covers the problem with having too few comments and too many comments.

Dr. Sorber listed a really interesting Truth about bad code and comments. What was that Truth?

Comments are in your program but do not get compiled or executed.

Why does Dr. Sorber believe comments exist?

Comments exist to make code.

Why do they not exist?

They don't exist because they compensate for bad code style.

What did Dr. Sorber state was your first priority when writing code and what makes good code.

The first priority is to write code that is easy to understand and has a coherent flow that doesn't repeat itself.

Dr. Sorber states having a lot of comments is not a bad thing but not always necessary. He discussed problems with requiring big comment blocks, what did he discuss?

It could often help with different systems for auto generating documentation. He discusses the problem with really heavy commenting is making sure that it stays up to date and how that becomes something that you have to maintain. If your comments get out of sync with your code, and you change code but you don't update the comments, the comments are worse than not having them at all. The person reading your code is going to think it's doing one thing when it does something else.

Code Smell's

## Segment 5 – Long functions

<https://www.youtube.com/watch?v=ll4XT0MYKN0>

Discuss the problem with really long functions.

The problem with really long function is that because they are so long, you can't keep them all on the screen at once; You can't keep them in your head at all times.

In Dr. Sorber's opinion, what is the rule of thumb with respect to the size of a function?

The rule of thumb is that you really want to keep your functions short enough so they can fit all on your screen all at once.

Dr. Sorber discussed another rule that could help keep your functions short and simple. What is that rule?

Another rule is having a function that is short and well named.