Homework
Due: August 24, 2021
You will submit this assignment 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.

These videos are short but very informative.

Code Smells

Segment 1 – Smelly code and Magic Numbers.
https://www.youtube.com/watch?v=p8RC_i9t0MU

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

- A magic number is just a numerical constant that should be replaced with a symbolic representation. It is essentially just a number on its own in the code.

Can you think of an example of a magic number?

- An example magic number would be like having 9.8 in an if statement when really you want to know if something is greater than acceleration to Earth's gravity:
    - If (acceleration > 9.8)
    - The above statement would be better as having had 9.8 defined and then incorporated in the if statement like:
        - Float EarthGravity = 9.8;
        - If (acceleration > EarthGravity) …

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

- The first problem is that while the number may work in the code, the person reading the code or even writing it later won't understand why it worked or why the number is there. The second problem with magic numbers is that it makes changing values harder later down the road. Say we have the same number appearing in multiple places, it would be easier to just change it in one spot where a variable is defined rather than finding every use of the magic number throughout the code.

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

- One example that Dr. Sorber gives that is acceptable to use a magic number is when you are using 0 to set the starting point for a for loop.

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?

- You should consider whether the elimination of magic numbers will make your code easier to maintain and easier to read in the future.

Code Smells

Segment 2 – Duplicate Code
https://www.youtube.com/watch?v=ck_RfVOYgjQ

Why is duplicate code not your friend?

- It is not good for a couple reasons. First, it makes your code longer, which could lead to having more bugs. Second, it makes your code harder to change due to the repeated logic.

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.?

- DRY stands for Don’t Repeat Yourself.
- His solution for repeated code is to make a function for that repeated code. Instead of repeating all the code over and over, you can just call the function.

He also states, the D.R.Y. principle makes your code _shorter__, easier to __read__ and __easier to understand____and __easier to debug and fix____.

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 where some of the issues that may arise for poor naming convention.

- Code can become hard to follow and read. It can also then become hard for us and others to go through and maintain or fix code later.

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.

1. Helpful – Example: int queueLength
2. Useful – Example: int BuildingHeight
3. Long/verbose - Example: int TheHeightOfTheBuildingInFeet
4. Lazy – Example: int a
5. Comical – Example: int blahblah

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

- You shouldn't be concerned with naming conventions when you are just writing something like silly example code
- You should be concerned when it is code that other people are going to read or other people are going to have to use and maintain it.

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.

1. Understandable – tells the reader what the thing is and its use
2. Consistent – you use naming conventions in the same way throughout your code (i.e name all class definitions in camel case)
3. Short – the name of the variable is short itself (fewer characters) to be easier to read/type

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

- Snake case – example: number_of_pens
- Camel case – Example: NumberOfPens
- Screaming snake case – Example: NUMBER_OF_PENS

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

- I use a mix of screaming snake case and camel case. For most instance variables I use camel case, but for constants I will use screaming snake case to make them stand out more.

Code Smells

Segment 4 Comments

DISCLAIMER: While I agree with Dr. Sorber in most instances, I have a slightly difference of opinion on comments. However, this video is a good video, hence, I am still assigning it.

I believe comments, even long detailed comment blocks are useful. Most of the companies you work for will have specific procedures they expect you to follow with respect to commenting. In this class I have a specific style of commenting I expect you to use in your program. These procedures will be given in the assignment documentation.

https://www.youtube.com/watch?v=LLqDNjr0kPo
This video covers the problem with having two few comments and to many comments.

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

- The Truth was that bad code with comments is still bad code. The comments still need to be useful.

Why does Dr. Sorber believe comments exist?

- They exist to make code more readable.

Why do they not exist?

- They do not exist to compensate for bad coding style.

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

- The first priority should be to write code that is easy to understand (i.e is well named, easy to follow, has good flow, no repetition).

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?

- He discusses that big comment blocks may become out of date and need to be maintained along with the code. So, if they don't stay updated it can make the code confusing to read later down the road.

Code Smell's
Segment 5 – Long functions
https://www.youtube.com/watch?v=ll4XT0MYKN0

Discuss the problem with really long functions.

- Really long functions can't be on the screen all at once and they can't be kept in your head at the same time either. So, you must scroll and search a lot, which makes bug searching and code fixing take longer and be harder.

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

- A function should be short enough to fit on your screen all at once.

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

- They should be simple enough that you can summarize what they do in one sentence.