Cs225 Exploring C++ Junseok Yang

Summary

1. (c++2019) The main theme of the first lecture is that the cpp is great. It is also said to focus on the essence when using C++ and only use advanced things when necessary. Also, when teaching c++, focus on the essence and don't hide the important points and important skills in the vast amount of information. I heard how the cpp was made from a historical perspective in the lecture. I also heard how the cpp works. In the lecture, the instructor also talked about the objectives of the cpp, saying it should be stable and provide adaptability. In the lecture, the instructor keeps saying that the cpp is good. But I didn't understand what the obvious topic was. In the lecture, the instructor explains the law of thumbs, which he says is important to the cpp. It is explained by ten laws. The instructor explains the law one by one. The conclusion of the lecture ends with a brief introduction to c++20.

2. (c++2018) The key to the Cppcon2018 lecture was how to make efficient code easily. He explained, showing eight examples of what is a simple code. They also say what is the best simple code. The instructor answers various questions, first of all, is the simple code. The answer to this is generally not. Second, simple code is correct. Third, simple code is not fast. The instructor also said that in order to make simple code as good as possible, you should not use a magic number, but name a variable as possible. Also, do not write long parameters of the function. And I recommend attaching const to the local variables we make." And the instructor was up six questions to the audience to answer. Also, the instructor will tell you how easy it is to use the tool. I took this lecture and learned how to use the simpler code and how to improve the quality of the code.

A summary of the topic

1. (c++2019) This summary topic is cpp is great, using C++ and only use advanced things when necessary. when teaching c++, focus on the essence and don't hide the important points and important skills in the vast amount of information.

2. (c++2018) This summary topic is how to make efficient code easily.

What is the point the speaker is trying to make?

1. (c++2019) Point is key c++ “Rules of thumb”

2. (c++2018) Point is in cpp simpler code. And more correct.

What are some of the motivations for the speakers point?

1. (c++2019)A static type system with equal support for program. And direct use of machine and operating system resources.

2. (c++2018) In this lecture, they are focusing on how we can create simple and efficient codes when we make them.

What are some examples?

1. (c++2019)In Coroutines : Better generators and pipelines

Lazy evaluation

Int main()

{

Auto src = seq(2);

Auto s = sieve(src);

Auto t = take(s,10,000);

Printf(t);

}

Coroutines : a bit of boilerplate

Generator<int> seq(int start)

{

While(true)

Co\_yield start++;

}

Generator<int> take(generator<int>& src, int count)

{

If(count <= 0)

Co\_return;

For(auto v : src) {

Co\_yield v;

If(--count == 0)

Break; }

}

2. (c++2018) Run fast simper code(usually, not fast)

for(auto p : people)

for(auto& p : people)

Was the talk interesting? Why or why not?

1. (c++2019) This is not interesting. Because, the instructor's lecture didn't make any sense. Also, it is difficult to know the subject matter in a lecture, and it is difficult to understand the contents of a lecture.

2. (c++2018) This lecture was interesting. Because it speaks the simpler code which is the key to the lecture. And in the middle of the lecture there was a painting or musical element to avoid boredom.

What did you learn from watching the video?

1. (c++2019) In this lecture, I learned how to use Libraries in "Rules of Thumb" efficiently. These can be used to create more efficient code.

2. (c++2018) Watching this lecture, I learned how to make the simpler code efficient.

How could you use it in your own code?

1. (c++2019) Use the added std library in my code to make a more efficient function.

2.(c++2018) Use header algorithm to make my code a simpler code. And not use the magic number.

Summary Demo(The End of std::endl)

<https://www.youtube.com/watch?v=6WeEMlmrfOI&list=PLHTh1InhhwT55y4fRRTBIelxnRSZ8G5yg&index=13>

(Demo) In the lecture, the instructor compares std::endl to '\n'. The theme of this lecture is that std::endl is slow. The instructor says he doesn't think of it as a big problem The average programmer uses std::endl to create a new line. std::endl also creates a new line, but it takes a little longer clearly than '\n' because it receives buffer information. std::endl is sometimes interpreted as '\n' + flush or '\n' ->'\n\r' but is usually interpreted as '\n' + flush'. I warn the audience not to use std::endl because the program slows down at the end of the lecture.

A summary of the topic

(Demo) This summary topic compares std::endl to '\n' std::endl is slow .

What is the point the speaker is trying to make?

(Demo) This speaker point is programmers use slow std::endl.

What are some of the motivations for the speakers point?

(Demo) This instruction speakers motivations are ‘\n’ is better than std::endl. Because, ‘\n’ is faster than std::endl. So speaker said can not using std::endl.

What are some examples?

(Demo) Experiment to generate 80 characters with 1,000,000 new lines with '\n' and std::endl. The experiment showed that '\n' took 0.5 s, and std::endl took 3 s.

Why was the talk interesting?

(Demo) I didn't know the contents of this course. So, taking this course was interesting to know that std::endl was lessening longer. I also got good information to do better programming.

What did you learn from watching the video?

(Demo) I found out that std::endl was slower than '\n' while I was looking at this course. Also, we found out that std::endl gives us the ability to create new lines and even flush.

How could you use it in your own code?

(Demo) When programming my code, replace it with '\n' rather than std::endl if possible for the speed of the program.

Create a demo that showcases the speakers point.

(Demo) I made my demo using <chrono>