**Pre-screening questions for C++ developers** 

1. What are your main achievements on your previous positions that you’re proud of?
2. What operating system do you develop on?
3. What do you use as your primary desktop OS for daily tasks?
4. What IDE do you use?
5. What database engines are you familiar with?
6. Do you use templates and metaprogramming?
7. What aredrawbacks of using templates and metaprogramming?
8. Which C++ 11 feature do you use most? Why?
9. Which C++ 14-20 feature do you use most? Why?
10. In which cases would you prefer using *class* instead of *struct*?
11. In which cases would you prefer using *enum* instead of *enum class*?
12. In which cases would you prefer using *const* instead of *constexpr*?
13. In which cases would you prefer using *typedef* instead of *using*?
14. When do you find using macro appropriate?
15. What could curly braces mean in C++?
16. What will be the output of this code snippet:   
       
    auto sum = [](auto a, auto b) { std::cout << a << " + " << b << " = " << (a + b); };   
    int i = 0;   
    sum(i += 2, i += 3);
17. Is it okay to throw exceptions in constructors or destructors?
18. Which synchronization methods do you use in multi-threaded code and when: mutex, condition variable or both?
19. What can be reasons to use *int32\_t* instead of *int*?
20. Write a simple method that prints the given enum class value as int?
21. Is it possible to inherit *enum class* from, say, an integer primitive type?
22. When is *git merge* more suitable than *git rebase?*
23. Have you been involved with any open-source projects?
24. Can you provide source code samples you consider to be good code (and are proud of)?
25. Which book has influenced you most (you would recommend it to each developer)?