C++ Notes to read from GFG

**Object Oriented Programming(OOP)**

1. [Object oriented design](https://www.geeksforgeeks.org/oops-object-oriented-design/)
2. [Introduction to OOP in C++](http://www.geeksforgeeks.org/basic-concepts-of-object-oriented-programming-using-c/)
3. [Classes and Objects](http://www.geeksforgeeks.org/c-classes-and-objects/)
4. [Access Modifiers](http://www.geeksforgeeks.org/access-modifiers-in-c/)
5. [Inheritance](http://www.geeksforgeeks.org/inheritance-in-c/)
6. [Polymorphism](http://www.geeksforgeeks.org/polymorphism-in-c/)
7. [Encapsulation](http://www.geeksforgeeks.org/encapsulation-in-c/)
8. [Data Abstraction](http://www.geeksforgeeks.org/abstraction-in-c/)
9. [Structure vs class](http://www.geeksforgeeks.org/g-fact-76/)
10. [Can a C++ class have an object of self type?](http://www.geeksforgeeks.org/can-a-c-class-have-an-object-of-self-type/)
11. [Why is the size of an empty class not zero?](http://www.geeksforgeeks.org/why-is-the-size-of-an-empty-class-not-zero-in-c/)
12. [Static data members in C++](http://www.geeksforgeeks.org/stati/)
13. [Some interesting facts about static member functions](http://www.geeksforgeeks.org/some-interesting-facts-about-static-member-functions-in-c/)
14. [Friend class and function](http://geeksquiz.com/friend-class-function-cpp/)
15. [Local Class](http://www.geeksforgeeks.org/local-class-in-c/)
16. [Nested Classes](http://www.geeksforgeeks.org/nested-classes-in-c/)
17. [Simulating final class](http://www.geeksforgeeks.org/simulating-final-class-in-c/)

**Dynamic memory allocation**

1. [new and delete operator in C++](http://www.geeksforgeeks.org/new-and-delete-operators-in-cpp-for-dynamic-memory/)
2. [malloc() vs new](http://www.geeksforgeeks.org/malloc-vs-new/)
3. [delete() and free()](http://www.geeksforgeeks.org/g-fact-30/)
4. [Pointers vs References in C++](http://www.geeksforgeeks.org/pointers-vs-references-cpp/)

**Virtual Functions**

1. [Virtual Functions and Runtime Polymorphism](http://www.geeksforgeeks.org/virtual-functions-and-runtime-polymorphism-in-c-set-1-introduction/)
2. [Default arguments and virtual function](http://www.geeksforgeeks.org/output-of-c-program-set-12-2/)
3. [Virtual functions in derived classes](http://www.geeksforgeeks.org/g-fact-36/)
4. [Can static functions be virtual?](http://www.geeksforgeeks.org/g-fact-29/)
5. [Virtual Destructor](http://www.geeksforgeeks.org/g-fact-37/)
6. [Virtual Constructor](http://www.geeksforgeeks.org/advanced-c-virtual-constructor/)
7. [Virtual Copy Constructor](http://www.geeksforgeeks.org/advanced-c-virtual-copy-constructor/)
8. [RTTI (Run-time type information)](http://www.geeksforgeeks.org/g-fact-33/)
9. [Can virtual functions be private?](http://www.geeksforgeeks.org/can-virtual-functions-be-private-in-c/)
10. [Inline virtual function](http://www.geeksforgeeks.org/inline-virtual-function/)
11. [Pure Virtual Functions and Abstract Classes](http://geeksquiz.com/pure-virtual-functions-and-abstract-classes/)
12. [Pure virtual destructor](http://www.geeksforgeeks.org/pure-virtual-destructor-c/)

**Exception Handling**

1. [Exception Handling Basics](http://www.geeksforgeeks.org/exception-handling-c/)
2. [Stack Unwinding](http://www.geeksforgeeks.org/stack-unwinding-in-c/)
3. [Catching base and derived classes as exceptions](http://www.geeksforgeeks.org/g-fact-60/)
4. [Catch block and type conversion](http://www.geeksforgeeks.org/catch-block-and-type-conversion-in-c/)
5. [Exception handling and object destruction](http://www.geeksforgeeks.org/exception-handling-and-object-destruction-in-c-set-1/)

**Namespace**

1. [Namespace in C++ | Set 1 (Introduction)](http://www.geeksforgeeks.org/namespace-in-c/)
2. [Set 2 (Extending namespace and  
   Unnamed namespace)](http://www.geeksforgeeks.org/namespace-in-c-set-2-extending-namespace-and-unnamed-namespace/)
3. [Namespace in C++ | Set 3 (Accessing,  
   creating header, nesting and aliasing)](http://www.geeksforgeeks.org/namespace-c-set-3-creating-header-nesting-aliasing-accessing/)