This note is a set of tips and tricks that I have learnt in coding with the C++ programming language.

RAII

RAII stands for *Resource Acquisition Is Initialization*. It is officially defined as “a C++ programming technique which binds the life cycle of a resource that must be acquired before use to the lifetime of an object. RAII guarantees that the resource is available to any function that may access the object. It also guarantees that all resources are released when the lifetime of their controlling object ends.” This briefly means that we need to:

1. Encapsulate each resource into a class: the constructor will acquire the resource, and the destructor releases the resource
2. Use the resource via a local instance of the class (i.e., not a pointer or static variable)

Let look at the following example.

Although function `notUseRAII` looks good, there is a potential memory leak. If an exception occurs while `doing something`, then `delete` is not invoked. Thus, there will be a memory leak.

Now, let apply RAII by wrapping `msg` using a `struct`.

As `wrapper` is a local variable, its destructor is guaranteed to be invoked when leaving function `useRAII` for any reason. Thus, the memory of the `msg` is ensured to be deallocated.

Ref: <https://en.cppreference.com/w/cpp/language/raii>

Working with containers

1. For range-based for loop
   1. Ref and copy
2. Std::find\_if
3. Std::distance