A Tour of C++ Book Report

In the book “A tour of C++”, the author Bjarne Stroustrup discusses many topics related to the programming language. From a high-level Bjarne tries to show many of the important features of the language as well as their purpose while not going too far in depth. I think the reason he does this is to make sure his audience, beginner to intermediate level C++ programmers, does not get confused but also walks away with a solid foundation of the language.

While C++ has a wide variety of features, libraries, frameworks, etc., the book aims to cover “most major language features and the major standard-library components” (S Stroustrup, Bjarne. “Preface.” *A Tour of C++*, Addison-Wesley, 2013, p. 181.). As stated earlier, this make Bjarne’s writings easy to follow. Of these features and standard-library components, the author’s overview of classes (concrete and abstract) and modularity (specifically namespaces) are the topics that I learned the most about while reading.

Up until recently, I never knew that classes could be split up into 2 categories (concrete and abstract). In the words of Bjarne, “the basic idea of a concrete class is that they behave just like built in types”(Stroustrup, Bjarne. “User-Defined Types.” *A Tour of C++*, Addison-Wesley, 2013, p. 181.). In other words, when you use a built-in type (i.e. int, float, etc.) or create variable in both cases you can utilize the operations/functions that come with that type. When you initialize an object of a concrete class, like a built-in type, you create an instance of that class that can have it’s functions and behaviors used. However with abstract classes, you’re only able to inherit its behaviors and functions via another class (sub class). Whether to use one or the other comes down to if the class you’re building has a unique implementation (concrete) or one that could possibly be reused with adjustments for specific use cases (abstract).

I didn’t know too much about namespaces before I started reading Bjarne’s book. However after reading his chapter on the topic, I did learn quite a bit to the point where I feel comfortable being able to utilize and explain their purpose. The purpose of namespaces is “expressing that some declarations belong together and that their names shouldn’t clash with other names” (Stroustrup, Bjarne. “Modularity.” *A Tour of C++*, Addison-Wesley, 2013, p. 181.) according to the author. Without namespaces we could run into scenarios where a program that contains thousands of lines of code can have a variable or object with the same name. This obviously will cause a compilation error as you cannot represent multiple objects or data types with the same name. To avoid this scenario in C++ you create a namespace that categorizes an assortment of names that should be used with one another. This namespace similarly behaves like a class or struct and when “declared” will allow users to name an object or data type with a namespace prefacing it.

This is the tip of the iceberg from what I learned in Bjarne's book. There are many more topics like error handling, containers, I/O streams, concurrency, and others that I learned while reading the book. I will be honest when I say that there are some topics that I will need to glance over again as I still do not understand them. However most topics were very well covered by Bjarne

and this in turn allowed to me to not only understand the syntactical intricacies of C++, but more importantly the many common “features” seen throughout programming languages and why and how they serve their purpose to programmers. This book set me on the right path to being able to utilize C++ effectively and learn other languages that share commonalities with it (specifically C#).