

- ✓ 2. Header Files
- ✓ 3. Using Headers with Multiple Files
- ✓ 4. Bjarne on Build Systems
- ✓ 5. CMake and Make
- ✓ 6. References
- ✓ 7. Pointers
- ✓ 8. Pointers Continued
- ✓ 9. Bjarne on pointers
- ✓ 10. References vs Pointers
- 11. Bjarne on References
- 12. Maps
- 13. Classes and Object-Oriented P...
- 14. Classes and OOP Continued
- 15. This Pointer
- 16. How Long Does it Take to Lear...
- 17. Outro



## References vs Pointers

Pointers and references can have similar use cases in C++. As seen previously, pointers can be used in pass-by-reference to a function. Additionally, they both provide a way to access an existing variable: pointers through the variable's address, and references through another name for that variable. But what are the differences between the two, and when should each be used? The following list summarizes some of the differences between pointers and references, as well as when each should be used:

References	Pointers
References must be initialized when they are declared. This means that a reference will always point to data that was intentionally assigned to it.	Pointers can be declared without being initialized, which is dangerous. If this happens mistakenly, the pointer will point to an arbitrary address in memory associated with that address, which could lead to undefined behavior and difficult-to-resolve bugs.
References can not be null. This means that a reference should point to meaningful data in the program.	Pointers can be null. In fact, if a pointer is null, it is often best practice to initialize it with a special type which indicates that the pointer does not point to any data.
When used in a function for pass-by-reference, the reference can be used just as a variable of the same type would be.	When used in a function for pass-by-reference, the pointer must be dereferenced in order to access the underlying data.

References are generally easier and safer than pointers. As a decent rule of thumb, references should be used in place of pointers when possible.

However, there are times when it is not possible to use references. One example is when you might like one object to store a reference to another object. However, if the reference is not available when the first object is created, then the first object will need to use a pointer to store the reference, since a reference cannot be null. The reference could only be initialized after the object is created.