

Introduction to Optional Values



Optional Values

- Sometimes it is desired to perform a task that doesn't have a valid result. Various ways of handling this...
 - A ***shamefully horrible*** technique is to use a *sentinel* value that indicates failure; e.g. -1
 - Return `true/false` for success, then use a reference parameter for the result
 - Use a `std::pair` with one value a boolean, the other the result
- None of these are needed anymore!!

Enter `std::optional`

- A helper class that optionally holds a value
 - header file: `<optional>`
- Various constructors to set (or not) a value
- Exposes `.has_value()` (`bool`) to indicate the presence of a value
 - Also overloads `operator bool`
- Exposes `.value()` that returns the contained value
- Easily explained through demonstration...



Code Demonstration – Optional Values

