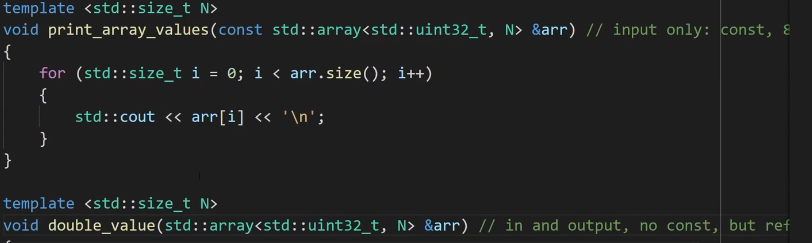
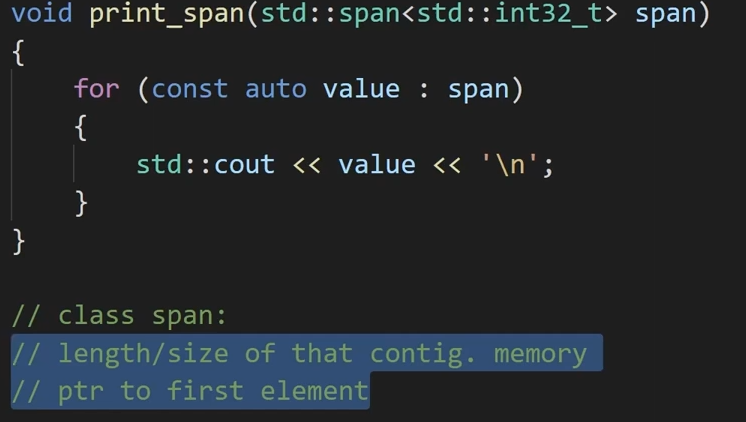
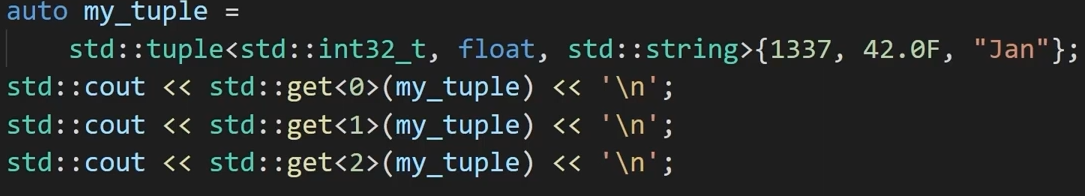
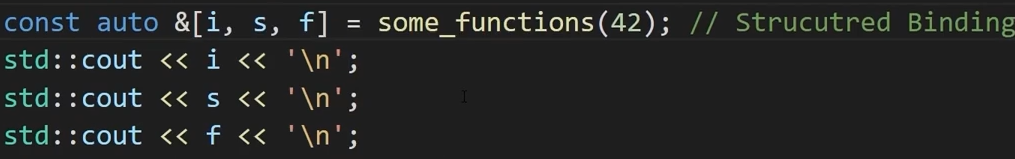
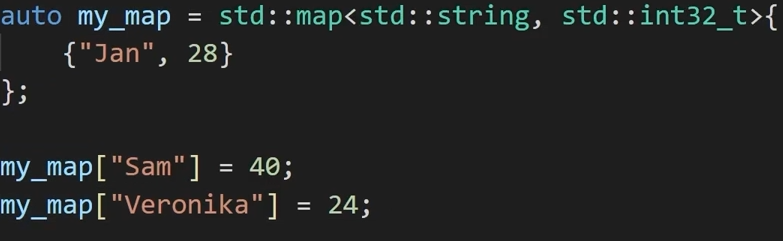
* **Structs:**
* Only use when I want to store data of different types, without the need of methods (if I need methods class is better)
* **Const:**
* Means the variable/object cannot be modified after initialization (Read only)
* The value can be set at runtime or compile time.
* **Constexpr**:
* Stronger guarantee: the value must be computable at compile time (when possible).
* Ensures the expression is a constant expression.
* Useful for array sizes, template parameters, switch cases, etc.
* **Anonymous namespace:**
* A modern C++ way to defining a function private to a single source file. (works like global variables) A screen shot of a computer code

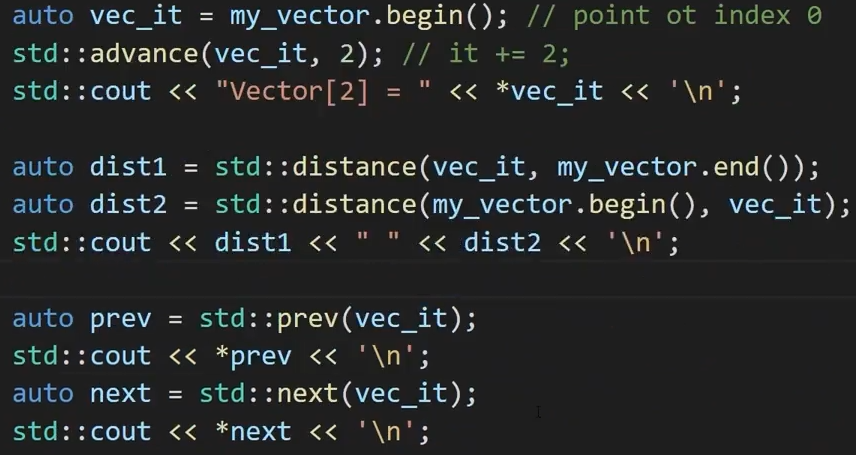
  AI-generated content may be incorrect.
* **Template:** is used to initialize arrays of size “N” (not a prefixed size)
* **Reference to array (pointer):**
* 
* Reference here is made in both cases but const for input only (read only) and in and output
* Use a pointer when you have to – otherwise use reference
* False of a pointer == **nullptr**, true of a pointer != nullptr (use in ifs)
* **lvalues**: An expression that refers to a specific object in memory (has an identifiable address).
* **rvalues**: A temporary value or a value that does not persist beyond the expression (no permanent address).
* **Templates:** Acts as Generics from Java:



**Span class:** does not own memory, it holds a reference to another container’s memory. So it can’t make a copy because it is a lightweight object which just holds a pointer to the first element of a (ANY – array, vector...) contiguous memory (an array/vector) 

* **Pair type:** Acts similar to a struct – define a pair of variables with type of choice (don’t have to be of same type) mainly used for map (pairs of key value)
* **Tuple type:** Acts similar to struct aswell. Defined like this: 
* **Structured binding:** alternative to “get “ for each variable in a pair or a tuple or struct or .... with a single line command: 
* A computer screen with text

  AI-generated content may be incorrect.
* **Map**: 
* Iterator commands that works for \*all\* containers in CPP



* **Strings:** Are mutable – can replace characters in strings.
* **std::string\_view** : when reading with “const std::string &s” as a function parameter – instead we should use std::string\_view – higher optimization.