Emplacement Solutions

emplace()

- What is the difference between insert() and emplace()?
  - When using insert() to add a new object to a container, a temporary object has to be created and copied into the container element
  - When using emplace() to add a new object to a container, the object is created directly in the container element
- What syntax is used for emplace?
  - Instead of passing the temporary object as the second argument, pass the arguments to the object's constructor
- Write a simple program which uses emplace() to add an object to a vector, then prints out the vector elements

emplace_back()

- What is the difference between push_back() and emplace_back()?
  - Similar to the difference between insert() and emplace()
  - When using push_back() to add a new object to a container, a temporary object has to be created and copied into the container element
  - When using emplace_back() to add a new object to a container, the object is created directly in the container element
- Write a simple program which uses emplace_back() to add an object to a vector, then prints out the vector elements

try_emplace()

- Briefly describe C++17's map::try_emplace()
  - map::try_emplace() checks for duplicates before creating any objects
  - The first argument to try_emplace is the new element's key
  - The remaining arguments are the arguments to the value's constructor
  - The returned value is the same as for insert()
  - If there already is an element with the same key, nothing happens
  - The "first" and "second" members of the new element are initialized by calling their constructors
- Why is map::try_emplace() needed?
  - map::emplace() always creates a temporary object, even if no element is added

try_emplace()

- Write a program which uses the try_emplace member function to insert a new element into an std::map
- Check your program works both for inserting a new element and assigning an existing element