

# Copying in Traditional C++ Solutions

- Describe some situations where the compiler adds copy operations in traditional C++
  - Passing argument by value
  - Returning by value
  - Populating standard containers
- Are there any ways these can safely be avoided?
  - Instead of passing by value, we can pass by const reference

- Why are these copy operations considered undesirable?
  - Copying creates a lot of overhead
  - e.g. vector of strings - allocate new memory buffer for vector, allocate new memory buffer for each string, copy data into each string, copy each string into vector
  - In situations where the copied object is about to be destroyed (returning from a function by value, inserting a local variable into a container) this overhead is wasteful and inefficient
  - Other languages do not have this problem (typically they use references which are managed by a garbage collector)