

Introduction to Smart Pointers Solutions

- Briefly describe the traditional C++ pointer
 - The traditional C++ pointer is a memory address
- Briefly explain why using traditional C++ pointers is undesirable
 - Too easy to make mistakes which cause program crashes (and worse)
 - Manual management of allocated memory complicates programs and is error-prone

- What is a "smart pointer"?
 - A smart pointer manages its own allocated memory
- Describe briefly how a smart pointer manages memory
 - Memory is allocated by the smart pointer's constructor and released in its destructor

- Why are smart pointers considered preferable to traditional pointers?
 - Avoid the need to write memory management code
 - The pointer is always valid
 - Encapsulation prevents the pointer being overwritten
 - No memory leaks

- In general, which pointer type is most useful?
 - `unique_ptr` for most cases where a pointer is needed
 - `shared_ptr` where multiple pointers to the same data would be used