

Smart Pointers Introduction Solutions

Traditional Pointers

- Briefly describe the traditional C++ pointer
 - The traditional C++ pointer is a memory address
- Briefly explain why using traditional C++ pointers is undesirable
 - Manual management of allocated memory complicates programs
 - Very easy to make mistakes which cause program crashes, data corruption etc

Smart Pointers

- What is a "smart pointer"?
 - A smart pointer manages heap allocated memory
- Describe briefly how a smart pointer manages memory
 - Memory is allocated by the smart pointer's constructor and released in its destructor
 - Special member functions can be used to transfer ownership of the memory from one smart pointer object to another

Smart Pointers

- Why are smart pointers considered preferable to traditional pointers?
 - Smart pointers avoid the need to write memory management code
 - The pointer always has an owner
 - Encapsulation prevents other code from interfering with the pointer
 - No memory leaks - the memory is always released, either at the end of a scope, or when an exception is thrown
 - Transfer of ownership is correctly handled - avoids need for "deep copying"