

# Classes and Traditional Pointers Solutions

- A class has a traditional pointer as a member. Describe how to write the constructor and destructor of the class (Hint: think about the RAII idiom)
  - We should allocate the pointer in the constructor and release it in the destructor

- What happens if we do not define a copy constructor or assignment operator for the class?
  - The compiler will generate default operators which perform a “shallow copy” by copying the pointer addresses
  - These will cause undefined behaviour if any copies or assignments are made
- Do these issues occur if the member is a smart pointer?
  - No. The default operators will call the smart pointer's copy operators

- Explain what is meant by the terms "shallow copy" and "deep copy"
  - A shallow copy means that the pointer addresses are copied
  - A deep copy means that the class acquires its own memory allocation which is correctly populated

- Describe how to implement a copy constructor which performs a deep copy
  - Allocate enough memory to hold the data from the other instance's memory allocation
  - Copy the data into it from the other instance's memory allocation

- Describe how to implement an assignment operator which performs a deep copy
  - If the two instances are identical (have the same address in memory), do nothing
  - Otherwise, release the memory allocation
  - Allocate enough memory to hold the data from the other instance's memory allocation
  - Copy the data into it from the other instance's memory allocation
  - Return the modified instance by reference

- What is meant by self-assignment?
  - Self-assignment means that an instance is assigned to itself
- Why is it a problem when we have a class with a traditional pointer as a member?
  - The assignment operator will delete the other class's buffer, then copy data from it. Undefined behaviour
- How can this be avoided?
  - Check whether the two instances have the same address

- Implement a class which has a traditional pointer as a member
- Your implementation must not cause any memory leaks or memory access errors
- Write a simple program which uses your class