C++

A S R Pavan
Scientist ‘B’
NIELIT Calicut

Topics to be discussed

- Smart Pointers
  - Issues with raw pointers
  - C++ smart pointers
    - Unique pointers
    - Shared pointers
    - Weak pointers

Issues with raw pointers

- Pointers provides absolute flexibility with memory management.
- Problems with pointers
  - Uninitialized pointers(wild pointers)
  - Memory leaks
  - Dangling pointers (pointer pointing to free memory)

C++ smart pointers

- Smart pointers can only point to heap allocated memory and they automatically called the delete when out of scope
- Defined by custom templates i.e., wrapped around raw pointers i.e., like objects only
- C++ smart pointers: code has no new's and no deletes
  - Unique pointers
  - Shared pointers
  - Weak pointers

C++ smart pointers

- Pointer arithmetic not supported(++,--,etc..)
-

Unique pointer

- Unique Pointer : ownership is unique
- It is not sharing the ownership
- Cannot be assigned or copied but can be moved
- So copy constructor is not available
- Syntax:
  - std::unique_ptr<int> ptr1 { new int {100} };
  - std::cout << *ptr1 << std::endl;
- reset method sets the ptr to nullptr: ptr1.reset()

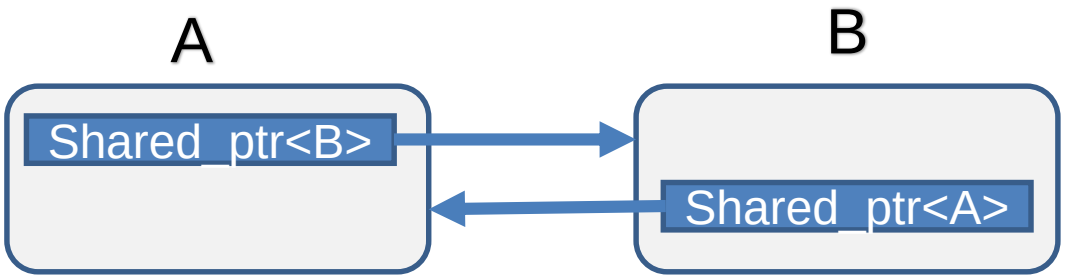
Shared pointers

- Provides shared ownership of heap objects
- Syntax:
  - shared_ptr<int> ptr1 {new int {100}}
- use_count method used to identify the ownership count

- make_unique
- Make_shared

Weak pointers

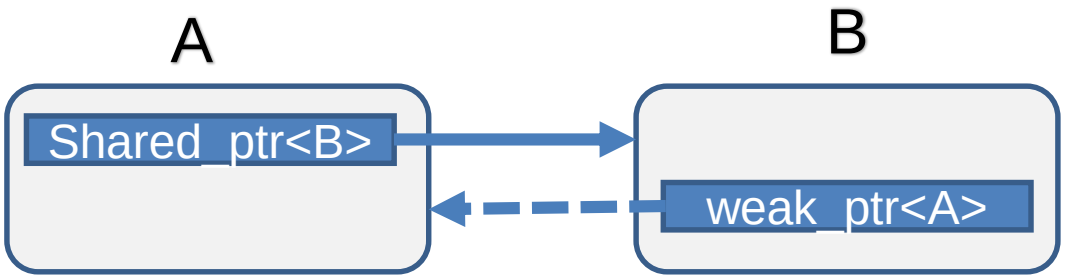
- Weak pointers won’t holding the ownership.
- Always created from shared pointer
- Doesn’t increment or decrement reference use count
- Avoids the problem associated with shared pointers (circular references)



- A has a shared pointer with B and viceversa

Weak pointers

- Weak_ptr : circular or cyclic reference



Doubts

Q&A

End of the session

Thank You