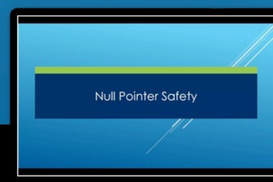
Null Pointer Safety

Making sure you are working with pointers containing valid memory addresses



```cpp
//Verbose nullptr check
int *p_number{};//Initialized to nullptr

if(!(p_number==nullptr)){
    std::cout << "p_number points to a VALID address : "<< p_number << std::endl;
}else{
    std::cout << "p_number points to an INVALID address." << std::endl;
}
```

```cpp
//Compact nullptr check
if(p_number){
    std::cout << "p_number points to a VALID address : "<< p_number << std::endl;
}else{
    std::cout << "p_number points to an INVALID address." << std::endl;
}
```

Calling delete on a pointer containing nullptr

```
//Calling delete on a nullptr is OK
int *p_number1 {};

delete p_number1;    // This won't cause any problem
                     //if p_number1 contains nullptr

//So no need to overdo something like
if(p_number1){
    delete p_number1;
    p_number1 = nullptr;
}
```