Static Members
- Variables – not part of the object, belong to the class, only one copy exist
- Functions – required only in declaration, belong to the class, do not receive this pointer, cannot access non static member functions, can be invoked directly through the class name

Constructors
- **Rule of 3** – All should be defined in a user implements any of them – **Destructor, Copy Constructor, Copy Assignment operator**
- **Delegating constructor –** constructor that invokes other constructor/constructors of the class

Copy of object
- **Shallow copy –** copy the address of pointer only
- **Deep copy –** allocate new address and then copy the value of the pointer

Rule of 5
- If class has ownership semantics, then you must provide a user-defined:
- **Destructor, Copy constructor, Copy assignment operator, Move constructor and Move assignment operator**

- **If any copy constructor/copy assignment operator or destructor is defined – move operations will not be created implicitly.**

- **if any move operation is defined – all other move operations and copy operations will be deleted**

Custom	Copy Constructor	Copy Assignment	Move Constructor	Move Assignment	Destructor
Copy constructor	Custom	=default	=delete	=delete	=default
Copy assignment	=default	Custom	=delete	=delete	=default
Move constructor	=delete	=delete	Custom	=delete	=default
Move assignment	=delete	=delete	=delete	Custom	=default
Destructor	=default	=default	=delete	=delete	Custom
None	=default	=default	=default	=default	=default

Copy/move elision
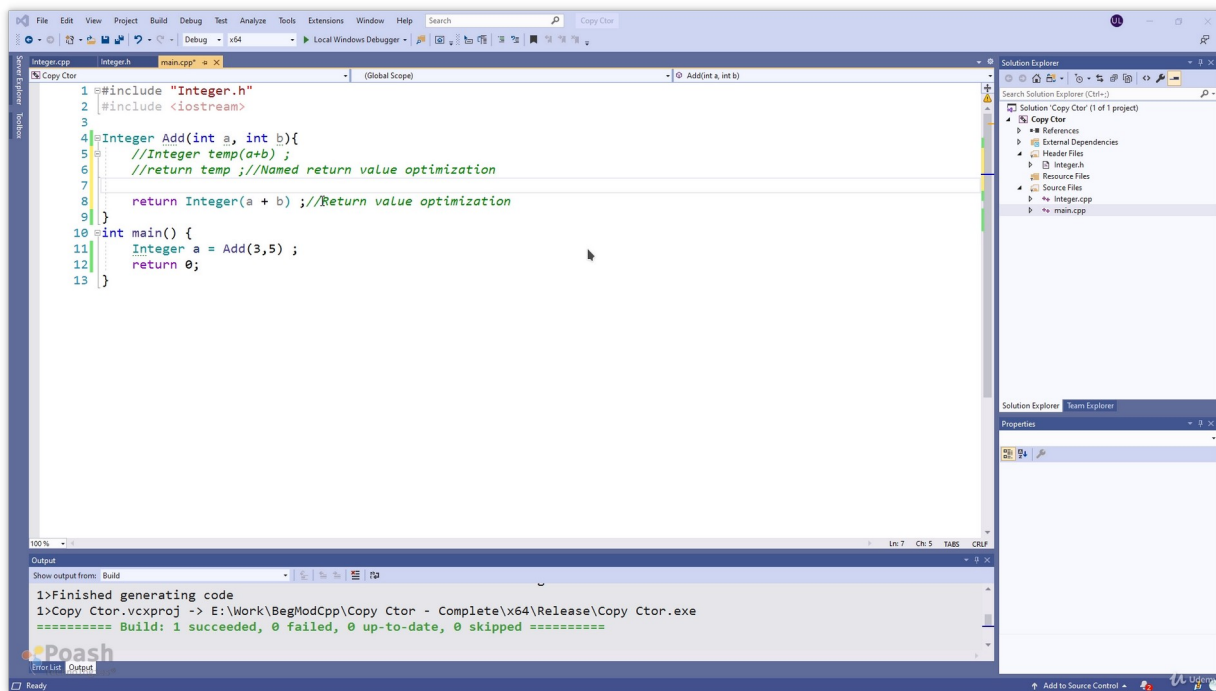
RAII principle – **resource acquisition is initialization** – The basic idea is to represent a resource by local object, so that the local object constructor will allocate the resource, and the local object destructor will release the resource. The problem is solved by smart pointers.
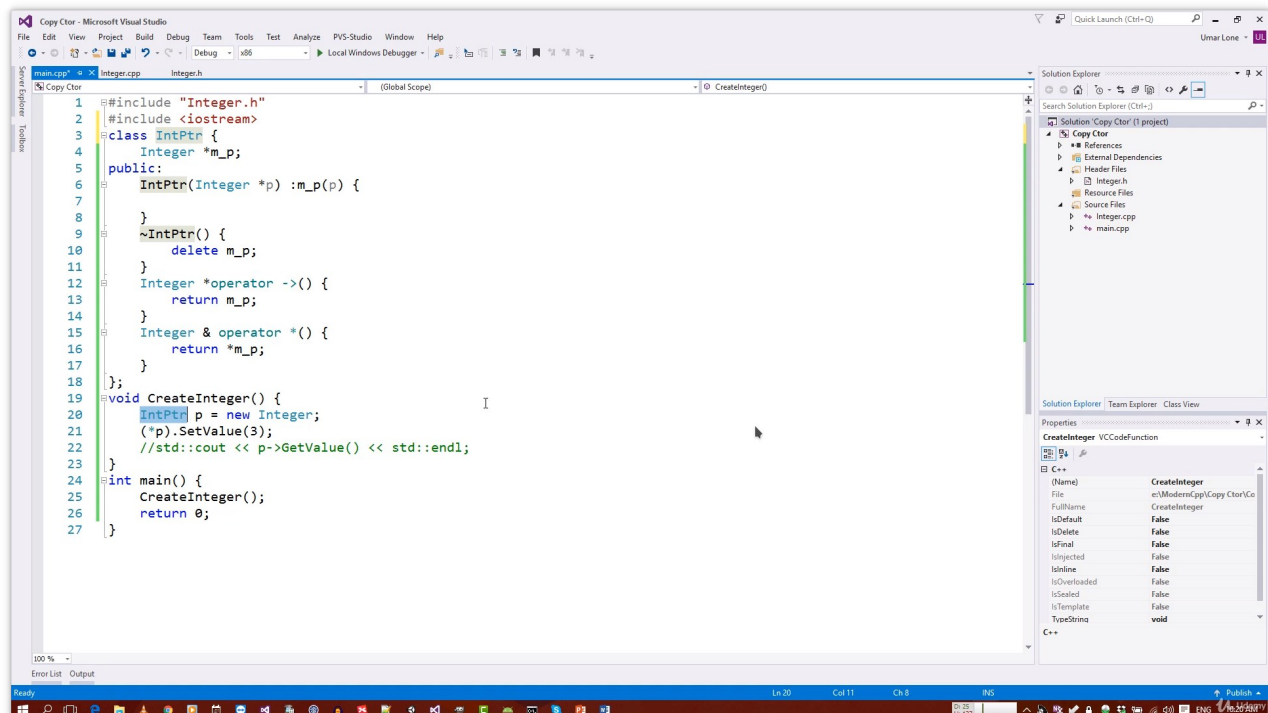
Type conversions

– Always prefer static_cast(c++ cast) over C-cast, because **C-cast do not check for the validness of the cast**.

- If we do not want constructor to be invoked implicitly due to type conversion (Integer a = 5), we have to **mark the constructor with keyword explicit**.

- Always prefer to use std::initializer_list to initialize members of the class. It allow initialization of member variables without assignment.