**Ray tracing in a weekend**

\r returns the cursor to the beginning of a line. So you can overwrite a line in console. Useful for progress updates.

----------------------------------------

Class members can be initialised in a member initialiser list rather than in the constructor body (more efficient). For example

Class myClass

{

Public:

myClass() : m\_int(1)

{}

Public:

Int m\_int

}

<https://stackoverflow.com/questions/751681/meaning-of-const-last-in-a-function-declaration-of-a-class>

https://stackoverflow.com/questions/455518/how-many-and-which-are-the-uses-of-const-in-c

const in a method declaration means you can implement two similar methods, one which is called when the object is const and one that isn’t.

**Rules and idioms for operator overloading**

<https://stackoverflow.com/questions/4421706/what-are-the-basic-rules-and-idioms-for-operator-overloading>

**Abstract classes**

https://www.ibm.com/docs/en/zos/2.4.0?topic=only-abstract-classes-c

Designed specifically to be used as a base class.

Contains one pure virtual function by using a pure specifier ( = 0 )

**Virtual functions**

Allow for dynamic function binding

<https://www.ibm.com/docs/en/zos/2.4.0?topic=only-virtual-functions-c>

**Shared\_ptr**

https://docs.microsoft.com/en-us/cpp/cpp/how-to-create-and-use-shared-ptr-instances?view=msvc-170

Smart points. Used when more than one owner might have to manage the lifetime of an object in memory. All instances of a shared ptr point to the same object with access to one “control block” that increments and decrements a reference count. When the reference count reaches zero the control block deletes the memory resource.

Use the **make\_shared** function to create a shared\_ptr.

Eg: auto eg1 = make\_shared<\*data\_type\*>(constructor)

Shared\_ptr is useful with C++ stl containers. If we wrap elements in a shared\_ptr then we can copy into other containers with the understanding that the memory is valid as long as you need it and no longer.

Shared pointers are used in the ray tracing series to allow multiple geometries to share a common instance (for example a bunch of spheres all use the same texture map material) which saves memory. It also makes memory management automatic.

**Iterators**

**Static keyword in method declaration**

Just like member variables of classes can ,be static, so can member functions of classes. Normal member functions of classes are always associated with a object of the class type. In contrast, static member functions of a class are not associated with any object of the class, i.e they have no \*this pointer.

Secondly since the static member functions of the class have no \*this pointer, they can be called using the class name and scope resolution operator in the main function (ClassName::functionName(); )

Thirdly static member functions of a class can only access static member variables of a class, since non-static member variables of a class must belong to a class object.

<https://www.learncpp.com/cpp-tutorial/static-member-functions/>

**This pointer**

The this pointer is a implicit pointer in an object that points to the object itself.

It is most commonly used to return a reference to the calling object, allowing for chaining of function calls on a single object

https://www.geeksforgeeks.org/this-pointer-in-c/