

Rule of Three

Things to Consider

- Proper memory use including use of delete
- Proper use of namespaces
- Do not use C functions such as:
 - strlen
 - strcpy
 - memcpy
- Prefer const to #define
- Proper use of struct vs data objects
- Use const instance functions - when function does not modify the object
 - <https://www.geeksforgeeks.org/const-member-functions-c/>
- Proper use of header files including the use of Inclusion guards (#ifndef or #pragma once)
- Do not break OO design approach with functions outside of a class that belong in a class
- Object oriented design - boundary classes

Rule of Three

(Geeks for Geeks)

if a class defines one or more of the following, it should probably explicitly define all three, which are:

- [destructor](#)
- [copy constructor](#)
- [copy assignment operator](#)

Default constructors and assignment operators only do “shallow copy”.

That is, the value of each data member is copied to the target in the case of an assignment copy. The implication of this is that any memory allocated explicitly by your class will not be copied. Only a reference to it.

This means that the memory allocated when the first object was created will be the memory to which the copy refers.

Code Example: RasterClass

Note:

- Constructors - when your class explicitly allocates memory on instantiation
- Destructor
- Assignment Constructor
- Not all pointers in your class need this treatment

Destructors are called when your application terminates. When your destructor is incorrect or incomplete, you may see error messages that include “core dumped”