Rule of 3 - Q&A

1. Content

2. Questions

Add your answers in the white boxes (in Dutch or English).

2.1. RuleOf3Basics

1) Crash details

In which function of the Container class happens the crash?
Destructor
Where in the Game class was this function called? (which function and where in that function, use the Call Stack window)
At the end of the "TestContainer" function
What happens there regarding the Container objects
It tries to delete a container object (c2) that is already deleted (c1)

2) Investigate the content of the variables just before the crash happens and draw your conclusions from this test.

Compare the content of both Container objects c1 and c2

Which data?	Where can you find it?
`m_Size`	Locals window
`m_Capacity`	Locals window
`Pointer to the dyn array: m_pElement`	Locals window
`Array elements: m_pElement,3`	Watch window

Do they have the same values? If not, which one are different?
They have the same values and m_pElement points to the same address
What can you conclude about the dynamic array in both Container objects?
They both get deleted when only one of them is deleted. Also, the pointers are dangeling.

Conclude: What happens by default when you create an object initializing it with another object of the same type?
The copy constructor gets called, executing a member-wise copy
Why does a crash happen when the containers c1 and c2 go out of scope?
The default copy constructor is called, using a shallow copy, which results in the pointer of both containers pointing to the same address

3) Changing an element in one of the containers.

What happens when you change an element in one of the 2 containers
The element in both the containers change
Why?
Both containers are pointing to the same address, thus changing the value at the address in one container updates the value of the first address in all the containers

4) Investigate the content of the variables related to the containers c1 and c2 just before they go out of scope.

Which data member(s) have the same value?
m_Size & m_Capacity
Which data member(s) have a different value?
m_pElement
Does changing the content of a container element, still influence the content of the elements in the other container?
No, because now both containers point to a different first address

5) Crash details

In which function of the Container class happens the crash?
The destructor
Where in the Game class was this function called? (which function and where in that function, use the Call Stack window)
At the end of the "TestContainer" function

6) Investigate the content of the variables just before the crash happens and draw your conclusions from this test.

Compare the content of both Container objects c1 and c3 Do they have the same values? If not, which one are different?
They have the same values and m_pElement points to the same address
What can you conclude about the dynamic array in both Container objects?
They both get deleted when one of them gets deleted

Conclude: What happens by default when you assign a Container object to another one ?
The copy assignment operator gets called, executing a shallow copy
Why does a crash happen when the containers go out of scope ?
The destructor for both containers get called, when the destructor on the last container gets called it crashes because the m_pElement pointer is already deleted

7) Investigation of what happens when an integer value is assigned to a Container object:
   c3 = 4;

Using the "Step Into" Debugger button, give a list of Container functions (only mention constructor, copy constructor, assignment operator or destructor) when this statement is executed. Write them down in order of execution. When the constructor is called also write down the value of the capacity parameter. **Don't mention the destructors** that are called when the 3 containers go out of scope at the closing curly brace of TestContainer.
Constructor, copy assignment operator, destructor
Constructor, copy assignment operator, destructor
4

8) Investigation of what happens when this code is executed:
   Container c4 = c1;

Only mention constructor, copy constructor, assignment operator or destructor when this statement is executed. Write them down in order of execution. **Don't mention the destructors** that are called when the 4 containers go out of scope at the closing curly brace of TestContainer.
Only the copy constructor is called to assign c1 to c4

9) Investigation of what happens when this code is executed:
   c4 = c2;

Only mention constructor, copy constructor, assignment operator or destructor when this statement is executed. Write them down in order of execution. **Don't mention the destructors** that are called when the 4 containers go out of scope at the closing curly brace of TestContainer.

Only the copy constructor is called to assign c2 to c4

10) Investigation of what happens when this code is executed:
c4 = CreateMultiplied(c1, 2);

Only mention constructor, copy constructor, assignment operator or destructor when this statement is executed. But also indicate why one was called. Write them down in order of execution. **Don't mention the destructors** that are called when the 4 containers go out of scope at the closing curly brace of TestContainer.

11) Investigation of what happens when this code is executed:
AddValues(c4, 3, 20, 30);

Only mention constructor, copy constructor, assignment operator or destructor when this statement is executed. Write them down in order of execution. **Don't mention the destructors** that are called when the 4 containers go out of scope at the closing curly brace of TestContainer.
Copy constructor 1, copy constructor 2, destructor, copy assignment, destructor

12) Creating a **static** Texture object

During the creation of a static Texture object something goes wrong with as consequence that it can't be drawn. When is the texture initialized? Why does the creation go wrong?
Before main is called. It cannot initialize static texture objects.

13) Assigning a Texture object to another one

When assigning a Texture object to another one, you get an error. Which deleted function are you trying to call?
The copy assignment operator

14) When passing a Texture object by value to a function

What deleted function is attempted to call
The copy constructor
How can you solve this error without changing the Texture class?
Change the texture param from pass by value to pass by const ref