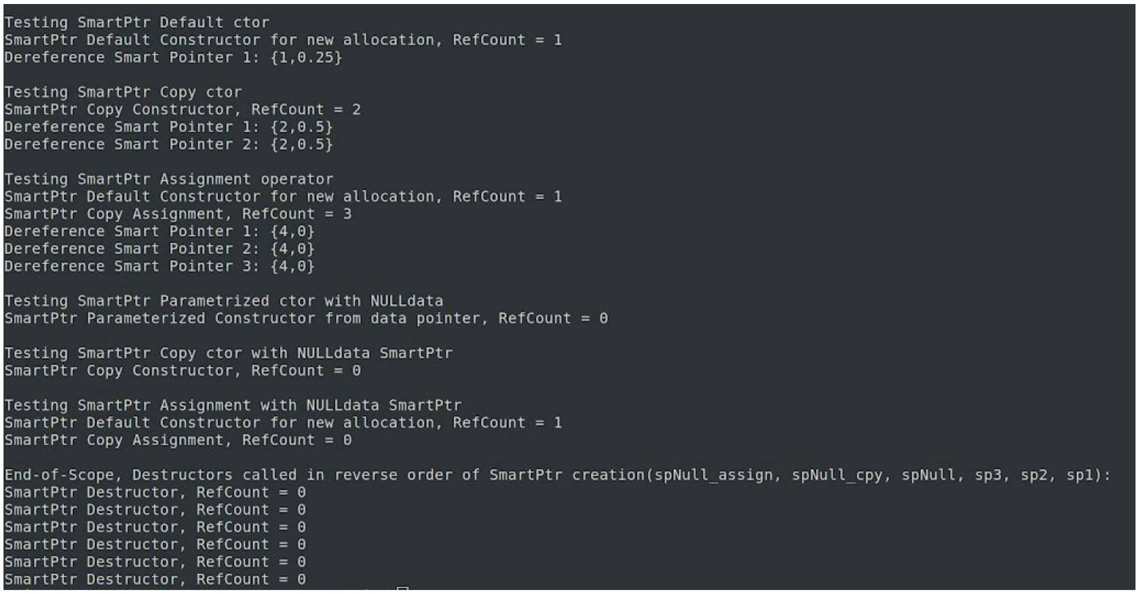
Prim Wandeevong
Lecturer Athanasia Katsila
Computer Science 202.1001
16 Sept 2019

Output



```
Testing SmartPtr Default ctor
SmartPtr Default Constructor for new allocation, RefCount = 1
Dereference Smart Pointer 1: {1,0.25}

Testing SmartPtr Copy ctor
SmartPtr Copy Constructor, RefCount = 2
Dereference Smart Pointer 1: {2,0.5}
Dereference Smart Pointer 2: {2,0.5}

Testing SmartPtr Assignment operator
SmartPtr Default Constructor for new allocation, RefCount = 1
SmartPtr Copy Assignment, RefCount = 3
Dereference Smart Pointer 1: {4,0}
Dereference Smart Pointer 2: {4,0}
Dereference Smart Pointer 3: {4,0}

Testing SmartPtr Parametrized ctor with NULLdata
SmartPtr Parameterized Constructor from data pointer, RefCount = 0

Testing SmartPtr Copy ctor with NULLdata SmartPtr
SmartPtr Copy Constructor, RefCount = 0

Testing SmartPtr Assignment with NULLdata SmartPtr
SmartPtr Default Constructor for new allocation, RefCount = 1
SmartPtr Copy Assignment, RefCount = 0

End-of-Scope, Destructors called in reverse order of SmartPtr creation(spNull_assign, spNull_cpy, spNull, sp3, sp2, sp1):
SmartPtr Destructor, RefCount = 0
SmartPtr Destructor, RefCount = 0
SmartPtr Destructor, RefCount = 0
SmartPtr Destructor, RefCount = 0
SmartPtr Destructor, RefCount = 0
SmartPtr Destructor, RefCount = 0
```

The default constructor is called for sp1.
The value of sp1’s m_refcount is 1. Sp1 is given values via “DataType.h” functions.
The operator* overload is tested on sp1, printing out sp1’s data.

The copy constructor is called for sp2.
The value of sp1’s m_refcount is increased to 2. Sp1’s data is copied to sp2.
The operator* overload is tested on sp1 and sp2, printing both of their data. It is the same because of the copy constructor.

The default constructor is called for sp3.
The assignment operator overload is used to make sp3 equal to sp1.
Since the number of SmartPtr objects that are now referring to sp1 is 3, m_refcount is equal to 3.
The operator* overload is testing on sp1, sp2, and sp3 to print all their data which is the same due to the copy constructor and assignment operator.

The parameterized constructor is called for spNull with the parameter “NULL” for m_ptr.
However the value of m_refcount does not increase because spNull’s m_ptr is null!

The copy constructor is called for spNull_cpy. The m_refcount still equals 0 because now spNull's m_ptr and spNull_cpy's m_ptr is null!
The default constructor is called for spNull_assign. By default m_ptr and m_refcount are not null and are allocated dynamic memory. Therefore m_refcount is equal to 1.

The assignment operator is called to make spNull_assign equal to spNull. Since spNull's m_ptr is null, spNull's m_refcount is 0. Therefore, spNull_assign's m_refcount is now 0 as well.

The program is ending so the destructors are called for each object. Since each object's lifetime is ending, the lifetime for m_refcount is also ending. Therefore, at the end of the program, m_refcount will be equal to 0.