Homework #5 – auto_ptr

In this homework you are asked to implement class auto_ptr. The following code shows the public interface you must implement:

```
template <typename X>
class auto_ptr
{
public:
   // Constructors
   explicit auto_ptr (X * = 0) throw();
   auto_ptr(auto_ptr &) throw();
   template <typename Y> auto_ptr(auto_ptr<Y> &) throw();

   // Destructor
   ~auto_ptr() throw();

   // Access
   X *get() const throw();
   X &operator*() const throw();
   X *operator->() const throw();

   // Assignment
   auto_ptr &operator=(auto_ptr &) throw();
   template <typename Y> auto_ptr &operator=(auto_ptr<Y> &) throw();

   // Release & Reset
   X *release() throw();
   void reset(X * = 0) throw();
};
```

Note that this interface is nearly identical to the auto_ptr interface defined in standard C++ (the only difference is that all functions working with auto_ptr_ref have been removed). Consult your standard C++ documentation for an explanation of how each function should work. **Be sure to provide unit tests for all code you implement.**

1. **(2 point)** Implement the constructors.
2. **(1 point)** Implement the destructor.
3. **(2 point)** Implement the access functions.
4. **(2 point)** Implement the assignment operators.
5. **(2 point)** Implement release & reset.
6. **(1 point)** Make sure your source code is well-commented, consistently formatted, uses no magic numbers/values, follows a consistent style, and is ANSI-compliant.

Place all source code and a screen capture of the output produced by your program in a single PDF document. Submit this document.