Enhanced Safe Array

Description

Implement a template class `EnhancedSafeArray`, which is a type of `SafeArray`. The class `SafeArray` implementation is given in *`safearray.h`*. Class `SafeArray` is a class that wraps a regular C++ array. It is considered a "safe" array class since it handles out of bounds array indexing through exception handling. Your class `EnhancedSafeArray` will augment the class `SafeArray` by supporting a copy constructor, a method to return the size of the array, an assignment operator, and an equality test operator.

Methods of class **`EnhancedSafeArray`**

- *`EnhancedSafeArray(void);`* This is the default constructor. It should invoke the default constructor of the parent class.
- *`EnhancedSafeArray(size_t);`*
  This is the single-parameter constructor. This should invoke the single-parameter constructor of the parent class.
- *`EnhancedSafeArray(const EnhancedSafeArray&);`*
  Copy constructor
- *`size_t size(void) const;`*
  Returns the number of elements stored
- *assignment operator ("=")*
  The result of this method must be that the calling `EnhancedSafeArray` must have equal size and elements as the argument `EnhancedSafeArray`.
- *equality operator ("==")*
  Two `EnhancedSafeArray` objects are equal if and only if their sizes and their stored elements are equal.

The prototypes above dictate the required method signatures. For the operators, devise appropriate signatures.

To complete this assessment, define the class `EnhancedSafeArray` in a file *`enhancedsafearray.h`*.

Files

Following is a list of files needed to complete this assessment.

- *`ex1.rar`* contains all of the following necessary files:
  - *`main.cpp`* - This file contains the main routine that tests your class `EnhancedSafeArray`.
  - *`safearray.h`* - This defines class `SafeArray`.

Tasks

To complete this assessment, you need to declare and define the template class `EnhancedSafeArray`.

To begin, verify the files needed for this assessment.

1. **Extract** the archive to retrieve the files needed to complete this assessment.

Following is an ordered list of steps that serves as a guide to completing this assessment. Work and
test incrementally. Save often.

2. **Begin** by creating the file *`enhancedsafearray.h`*. Place your declaration and definition of class `EnhancedSafeArray` in this file.
3. **Next**, declare class `EnhancedSafeArray`. Use inheritance appropriately to model the fact
that an `EnhancedSafeArray` is a type of `SafeArray`. Use the method signatures listed
above.
4. **Then**, implement the two constructors of class `EnhancedSafeArray`. Both of these
constructors should invoke the appropriate parent class constructor.
5. **Next**, implement the copy constructor. The copy constructor should make a deep copy of
the source `EnhancedSafeArray`. This involves allocating memory for the inherited
`storage` array and copying into it all of the elements in the source `EnhancedSafeArray`.
6. **Then**, implement method `size`. The method returns the number of elements in the `EnhancedSafeArray`.
7. **Next**, implement the assignment ("=") operator.
8. **Finally**, implement the equality ("==") operator.
9.

Submission

Submit **only** the following.

1. *`enhancedsafearray.h`* - finished version of class `EnhancedSafeArray`