## Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## CS142 Retest #1

1. Consider the Costume class. What member and non-member functions are invoked at each of the numbered statements? For each function that is invoked, specify the full name of the function and the number of times it is called.

Costume myFunction(Costume theCostume)

{

 return theCostume;

} // 8

1. If (myCostume == yourCostume)

 {

2. Costume myOtherCostume (“Scary Professor”);

3. Costume myNewCostume= myCostume;

4. Costume\* mostPopularPtr = new Costume(“Wicked Witch”);

5. myNewCostume = \*mostPopularPtr;

6. delete mostPopularPtr;

7. myOtherCostume = myFunction(myFavorite);

 }

 7.

1. bool Costume::operator==(const Costume& rhs) const // (called once).

2. Costume::Costume(const char\* costumeName); // (called once)

3. Costume::Costume(const Costume& orig); // (called once)

4. Costume::Costume(const char\* costumeName); // called once

5. const Costume& Costume::operator=(const Costume& rhs); // called once.

6 . Costume::~Costume(); // called once

7. Costume::Costume(const Costume& orig); // Copy constructor used to pass myFavorite by value

 Costume::Costume(const Costume& orig); // Copy constructor to return copy of theCostume

 const Costume& Costume::operator=(const Costume& rhs); // called once to copy result to myOtherCostume.

8. Costume::~Costume(); // called once

1. Consider the following program fragment:

char alphabet[] = “abcdefghijklmnopqrstuvwxyz”;

1. Write a statement that declares a new pointer that points to alphabet. The declaration should protect the letters in alphabet from being changed through the use of this pointer.
2. Write a statement that declares a new pointer that points to alphabet. The declaration should protect the pointer itself from being changed.
3. Write a statement that declares a new pointer that points to alphabet. The declaration should protect the letters in alphabet from being changed through the use of this pointer and should protect the pointer itself.
4. Write a statement that uses the declaration from part b to change the value of the 4th letter in alphabet to ‘x’.
5. Using one or more of the pointers declared in the previous questions, change all of the letters in alphabet to ‘x’.
6. Write a Costume class that includes the following public member functions:
	* Constructors
		+ Constructor that takes a costume name as a parameter
			- Parameters:
				* costume name (e.g., “Wicked Witch”)
		+ Copy Constructor
	* A Destructor
	* A function called addElement that adds an element to the costume.
		+ Parameters:
			- Element name (e.g., “scary mask”)
		+ Return Value: None
	* A function called getNumberOfElements
		+ Parameters: None
		+ Return value: The number of elements in the costume.
	* An assignment operator
		+ The assignment operator allows statements such as:
			- myCostume = yourCostume, where myCostume and yourCostume are instances of class Costume.
	* An equality operator
		+ The equality operator allows statements such as
			- If (myCostume == yourCostume), where myCostume and yourCostume are instances of class Costume
			- Costumes are considered equal if they have the same list of elements (added in the same order) and the same costume name.
	* A bracket operator that references the ith ingredient. The bracket operator must work on both const and nonconst objects.
		+ Parameters: an integer index
			- Example:

Costume myCostume (“Thomas the Tank Engine”);

myCostume.addElement (“Engineer Cap”);

myCostume.addElement (“Thomas Mask”);

myCostume.addElement(“Thomas Vest”);

cout << myCostume[1] << endl; // prints: Thomas Mask

myCostume[1] = “Coal”; // Changes second Element to “Coal”

* + Important Note: The Costume class must be implemented so that the **name of the costume** is a private data member of type char\*. You may use the string type elsewhere in the program, for example, for the list of elements.
	+ A function called getNumberOfCostumes that returns the number of Costume objects currently in existence. For extra credit, return the number of unique costumes instead.
	+ An insertion operator (<<) that allows instances of the Costume class to be printed as follows:

cout << myCostume << endl;

 this would print:

 Thomas the Tank Engine:

 Engineer Cap

 Thomas Mask

 Thomas Vest