1. What are the advantages of the *new* operator over the C library functions malloc, etc.?

The following answer assumes we are working in C++ only:

new is easier to use than malloc.

To allocate memory with new, the user needs just write new <data type>, and new automatically finds a block of memory of the correct size and return an address to that block.

When using malloc, the user needs to specify the size, which can be prone to errors. Also, the operator new returns a pointer of the correct type. Malloc will return a void pointer which the user needs to type cast to the correct type which can also be prone to errors.

1. How do you know which constructor is the copy constructor?

A copy constructor for a class has the following signature:

<Class name>(const <Class name> &)

where the argument is a constant reference to a class object.

1. How are the copy constructor and the overloaded assignment operator different?

* The overloaded assignment operator (=) must use the delete [] operator in the function definition since the target object may already refer to allocated data; and the function should free former obligations. In contrast the copy constructor is used for instantiation of new objects: it does not need to use the delete [] operator because the target object does not refer to allocated data.
* The overloaded assignment operator should protect against assigning an object to itself. The copy constructor does not need to do that because we are creating a new object.
* The overloaded assignment operator returns a reference to the invoking object while the copy constructor does not have a return type.

1. When does the copy constructor get used?

The compiler uses a copy constructor anytime the program generates a copy of an object. A copy constructor is used whenever a new object is created and initialized from an existing object of the same kind. For example: when a function passes an object by value, when a function returns an object or creating a new object form another one.

1. In the dynamic array example, we apply the subscript operator to something that isn't even an array. How is this possible?

* Because the subscript operator [] operator was overloaded to work with the user defined data type DynamicIntArray. When the user uses a subscript operator [] in a DynamicIntArray, the overloaded function applies the [] operator to the ‘data’ private data member which is an int array.