Name: Soham Thaker SID: 011-748-159 Email: [sdthaker1@myseneca.ca](mailto:sdthaker1@myseneca.ca)

In this workshop I learned how to apply rule of 3 to the classes in a hierarchy. The rule of 3 consists of a copy constructor, copy assignment operator and a destructor for each class a hierarchy.

To begin with, lets talk about copy constructor. A derived class with resources needs explicit definitions for its special member functions which are the rule of three. If a programmer explicitly writes a copy constructor for derived class, without a call to base class’ copy constructor, it will call the default argument constructor by default. In order for us to the copy constructor we must call the copy constructor from base class explicitly in the initializer area and pass the right operand received in the derived class copy constructor. By passing the right operand we do an upcast which always succeeds since every object of derived class is inherited from the base class. Also, by doing this, the copy constructor in derived class executes the base class constructor first and then executes the logic defined in the derived class constructor. Usually, if the logic of the copy constructor is same as the copy assignment, we can use the same logic defined in the copy assignment operator by calling the copy assignment in this way “\*this = Ro” which is an invocation to the copy assignment operator that belongs to the types of these two objects.

Furthermore, explicitly defined copy assignment of derived class does not automatically call the copy assignment from base class where we must call the copy assignment of base class within the scope of copy assignment of the derived class. The way we call the copy assignment operator for base class is we do a casting on reference of current object and assign it the Right operand that was received in the copy assignment of derived class. By doing this we are upcasting from derived class part of the object to the base class part of the object which always succeeds. Besides, there are precisely 5 steps to do copying for classes that have resources. The first step is to self assignment, followed by the shallow copying for non-resource variables, followed by deleting memory allocated for the resources in the current object, followed by performing a deep copy for the resources of the current object by allocating memory for the resources and then copying the data from the right operand’s resources to the current object’s resources and finally return the reference of the current object to update the values of the data members on the current object. If we are executing a copy assignment where a derived and base, both have resources, we usually do the deep copying for base class part of the current object first by calling the base class’ copy assignment operator explicitly and then take care of the derived class’ resources.

Moreover, destructors for classes in a hierarchy are called automatically by their objects at the time of object going out of scope where a derived class object calls the destructor from the base class automatically, once executed then executed its own destructor.