COMP 116 Data Structures

Lab #6

In this lab, we will practice the use of classes, methods and operator overloading. To do this, you will create a new type of object called `IntTriple`.

For all the exercises below, feel free to either write the class entirely in one file, or to split it between a .h and a .cpp file. Note however that separating between a .h and .cpp file is the usual way that these things are done, so it would be good practice to try it. You should also write a `main` function that allows you to test the features of the object you just programmed.

1. First, we need to know what kind of data the object contains. The class `IntTriple` should contain three integers and a boolean value `init` that will remember whether the `IntTriple` has been initialized or not. These data members should be `protected`. All methods below will be `public`.

2. Then we need the constructors and destructor. You should program:

   (a) A default constructor that does not initialize the integer members.

   (b) A constructor that takes three integers as parameters and uses them to initialize the three integer members of the class.

   (c) A copy constructor which, given an `IntTriple` as input copies all its data members.

   (d) A destructor that reclaims the resources allocated by the object (what are they?).

   For each of the constructors, you should set `init` to the appropriate value.

   ________ Show me the result when you are done.

3. Then we need a few methods

   (a) Write a method `isInitialized` that returns whether the `IntTriple` has been initialized.

   (b) Write methods `getFirst`, `getSecond` and `getThird` that return the first, second and third integer contained in the triple. These methods should throw a `runtime_error` if the `IntTriple` is not initialized.

   (c) Write a method `set` that takes three integer parameters, sets the three integer data members of the triple accordingly, and modifies `init` if needed.

   (d) Write a method `to_string` that returns a string containing the elements of the triple separated by commas, surrounded by a set of parentheses. For example, if the elements of the triple are 1, 5 and 8, the method `to_string` should return "(1,5,8)". The method should return "()" if the triple is uninitialized.

   ________ Show me the result when you are done.

4. Now, let's overload a few operators:
   (a) Overload the assignment operator (operator=) so that assigning an `IntTriple` copies all the data members and returns a reference to the assigned triple.
   (b) Overload the addition operator (operator+) so that adding two `IntTriple`s returns a new `IntTriple` whose members are the sum of the two operands. For example, $(1, 2, 3) + (5, 1, 2) = (1 + 5, 2 + 1, 3 + 2) = (6, 3, 5)$. The operator should throw a `runtime_error` if either operand is not initialized.
   (c) Overload the multiplication operator (operator*) so that multiplying two `IntTriple`s returns an integer containing the *dot product* of the two triples. For example, $(1, 2, 3) * (5, 1, 2) = 1 * 5 + 2 * 1 + 3 * 2 = 13$. The operator should throw a `runtime_error` if either operand is not initialized.

   ________ Show me the result when you are done.

5. Already all done? Let's go do a bit more then.
   (a) Overload the equality operator (operator==) to return `true` if either both `IntTriples` are not initialized, or if they are both initialized and all their respective integer data members are equal. The method returns `false` otherwise.
   (b) Overload the 'less than' operator (operator¡) to return `true` if any of the following is true:
      - the first integer of `this` is less than the first integer of the other operand.
      - the first integer of both `IntTriples` are equal, but the second integer of `this` is less than the second integer of the other operand.
      - the first two integers of both `IntTriples` are equal, but the third integer of `this` is less than the third integer of the other operand.
   (c) Overload the other comparison operators ($>, <=, >=, !=$) in the logical way given the two operators defined above.

   All these operators should throw a `runtime_error` if one of the operands is uninitialized.

   ________ Show me the result when you are done.

When you are done, write your name on the sheet and hand it to the lab instructor.

Name: ____