**CSIS 123A Lab Assignment 2 Class Construction**

Create a class called Integer and another called Double. These classes should have separate header and .cpp file

In order for you to be successful you will need to determine what the data section of the class consists of.  Each of these classes should have the following functions (methods)

**equals** - This is a void function that sets the value of the object. For example, an instance of Double should set a double value:

Double d;  
d.equals(12.34);

**add   
sub  
mul  
div**

Each of these functions should take its type as an argument and return the same. For instance, a Double should work on Doubles, an Integer should work on Integers

The Integer class should have a function called

**toInt()**

Which returns a primitive integer. Basically, this is the data section that gets returned.

The Double class should have a function called

**toDouble**

Which returns a primitive double

In your main function write code to test instance of your classes

[**Assignment 3 - Due Feb. 06**](https://my.msjc.edu/webapps/assignment/uploadAssignment?content_id=_3049638_1&course_id=_78102_1&assign_group_id=&mode=view)

**CSIS 123A Lab Assignment 3 Overloaded Constructors**

For your Double and Integer classes add a default constructor that sets the value to 0.0 and 0 respectively. Then add the following overloaded constructors

**Double class**

A Double argument

A primitive double

An Integer class

**Integer class**

An Integer class

A primitive int

Each of these constructors should set the data section of the class to the value being passed to it.

In addition to the overloaded constructors add the following overloaded functions

add

sub

mul

div

Each of these should take primitive values and return the base class type. For instance, a Double class should be like this:

Double d(12.50), d3;

d3 = d.add(1.5);

In this case d3 should now have a value of 14.00

**CSIS 123A Lab Assignment 4 Using Complex Types - Vectors**

In this lab you are going to use the menu you created in lab 1 to test both your Double and Integer classes. You should add functionality to the menu to allow you test the add, sub, mul, div functions for instances of both classes

You are going to have make a modification to your menu class. Currently it uses an array to hold a fixed amount of menu items. While this may be ok for most applications we want our menu to be as flexible as possible. To make this happen I want you to remove the array of menu items and replace it with a vector.

We want all of our code to be as organized as possible. You should have separate header and .cpp files for each class with your main function being in a separate file.

Because the menu is setup to take pointers to void functions you are going to have to create a layer of functionality that sits between the two. This kind of structure can be thought of as a three tier approach.

Here is a simplistic example

void doubleAdd();

int main()  
{

m.addMenu("1. Add Doubles", doubleAdd);  
m.runMenu();

}

void doubleAdd()  
{

 // Code to add two Double classes here. Should get input from the user  
m.waitKey();

}