# Object-Oriented Programming I Fall 2016

**CIS 9310**

## **Programming Assignment 3**

For this assignment you are to implement a class that will be used to instantiate objects that denote English lengths in the form of feet and inches. More specifically, objects of this class will be used to represent values that are depicted in the form *f’i”*. Examples of such values are: 1’8” (i.e., 1 foot, 8 inches), or 21’2.9”, 0’0”, etc. Note that feet is always denoted as an integer value, whereas the inches component may have fractional part. Your class implementation should adhere to the following specifications:

1. The identifier for this class should be **Length**.
2. The class should have the following include the following public interface:

### Constructors

**Length();**

Default constructor that initializes a length object to value *0’0”*.

**Length(int f, float i);**

Initializes object to value **f’i”.** The second parameter should have a default value of 0.

### Access Functions

**int getFeet();**

Returns feet component of a Length object.

**double getInches();**

Returns inches component of a Length object.

**double getLength();**

Return value of Length object expressed in inches.

**Length Add (Length l);**

Adds value of l to current object and returns the result. Does not change value of either operand – the object to which you are sending the **add** message, and the object denoted by the parameter **l**. E.g., the expression “**len1.add(len2)**” returns the sum of **len1** and **len2** without changing the values of **len1** and **len2**.

**bool LessThan(Length l);**

For two **Length** objects *l1* and *l2*, *l1.LessThan(l2)* returns *true* if *l1* is less than *l2*.

**bool GreaterThan(Length l);**

For two **Length** objects *l1* and *l2*, *l1.GreaterThan(l2)* returns *true* if *l1* is greater than *l2*.

**bool EqualTo(Length l);**

For two **Length** objects *l1* and *l2*, *l1.EqualTo(l2)* returns *true* if *l1* is equal to *l2*.

### Modifier Functions

**void setFeet(int f);**

Set value of feet component of Length object to f.

**void setInches(double i);**

Set value of inches component of Length object to i.

**void setValue(int f, double i);**

Set feet component of Length object to f and inches component to i;

**void AddTo(Length l);**

Adds value of l to current object. E.g., the expression “**len1.addto(len2)**” adds the value of **len2** to that of **len1**, modifying the value of **len1**.

### Input/Output Functions

**void Read();**

Reads a length value expressed in the form *f’i”*. (This function does not, and should not, provide a user prompt.)

**void Write();**

Outputs a length value in the form *f’i”*.

Using the **Length** class you are

Using this **Length** class you should implement a program that prompts for and reads three length values from a user, and then outputs them in ascending length order. You should also implement a main() function to perform a *complete* test of your **Length** class implementation. I may supply also a main() function to test your implementation. If I do, the output of this program should also be submitted along with the output generated by your main() functions.

Due Date: November 17, 2016