Lab 2 preview worksheet

26 January 2017

`Location` data design

1. Describe the data.

Based on the context from the lab handout, describe what `Location` itself will be responsible for.

2. Give examples.

Give a few examples of `Location`s. Make notes about why they are significant and/or interesting.

Example name	Example	Notes

`Location` method design: accessing the first coordinate

1. Describe the method.

Fill in the blank in the following description as appropriate:

> Returns the _____ of this `Location`.

2. Declare the method, define a stub.

> What type of value will it be returning, if any? _____
>
> Does it have any "given" values, and if so, what types? _____
>
> Will it modify "this" value? _____

Write out the method declaration (which will eventually go in `Location.h`):

And then the stub method (which will eventually go in `Location.cpp`):

3. Write test cases.

Write two expressions that make use of the method we're designing, and their expected results.

`check (` _____ `) expect` _____ `;`
(Expression to evaluate / Expected result)

`check (` _____ `) expect` _____ `;`
(Expression to evaluate / Expected result)

`Location` method design: `isEqualTo`

1. Describe the method. (I'll give you this one.)

Determines whether this `Location` is the same as a given other `Location`.

2. Declare the method, define a stub.

What type of value will it be returning, if any? _____

Does it have any "given" values, and if so, what types? _____

Will it modify "this" value? _____

Write out the method declaration (which will eventually go in `Location.h`):

And then the stub method (which will eventually go in `Location.cpp`):

3. Write test cases.

Write out an appropriate number of test cases.