CS132 Project1
2018 fall
streller

PROBLEM STATEMENT:

The Logo language, made the concept of turtle graphics famous. It was a mechanical turtle that walked around and drew shapes. The turtle holds a pen in one of two positions, up or down. While the pen is down the turtle traces out shapes as it moves; while the pen is up, the turtle moves about freely without writing anything. This is similar to the way some line draw programs work. You are to design such a program

CODE:

Use a 20 by 20 array or floor which is initialized to zeros. Keep track of the current position of the turtle at all times, the direction it is heading, and whether the pen is currently up or down. Assume that the turtle always starts position 0,0 on the floor with its pen up, heading to the right or east. The set of turtle commands your program must process are:

1	change pen position
2	turn
3	move forward
4	print
5	change brush
6	erase/start over
7	jump
8	see menu again
9	end program

command 1 is to act as a toggle
command 2 is to rotate 90 degrees from current direction (direction of rotation is user supplied)
command 3 is to move forward x units (x is user supplied)
command 4 will print/display the entire floor
command 5 will present four different character/brush options
command 7 will jump/move the turtle to a new position (x,y) on the floor, which are user supplied

Assuming the turtle is near the center, heading east the following would draw a 12x12 square

```
1
3  12
2  S
3  12
2  W
3  12
2  N
3  12
```

There must be at least

a turtle class (turtle.h , turtle.cpp)
a pen class (pen.h, pen.cpp)
a direction class (direction.h , direction.cpp),
a position class (positiion.h, position.cpp)
and a driver with the name cs132_P1_driver.cpp..

The driver must not contain any function prototypes/implementations: place them in separate files
It should be noted, that the position is not part of the turtle, hence the details should be hidden from the turtle (ie its own class, and separate implementation.) Likewise for direction..

When an object exhibits a " has a ..." relationship – called composition (of classes) in OOP – separate classes are used. For example, the turtle object has no special privileges of the position object. A turtle walks on the floor, ie uses the position class though its interface. Does the turtle need to know the size of the floor? The turtle uses the floor but that's about it. If an object exhibits a " is a .." relationship, then inheritance is used.

DELIVERABLES:

hard :In a bound folder

1. documented source code
2. user manual.
3. Programmer manual(s) (one for each class also)

soft: soft: a zipped file, called CS132_P1_yourLastName, containing

1. all source code
2. release version executable

submitted in Blackboard to CS132

Status Report Due 3 October 2018

emailed to streller@ecc.edu with the subject cs132_P1_statusReport

1. This **must be** a pdf.
2. the content will be a description of what you have completed as of 16 February. (even if you haven't begun the project, you must state so and still submit the report)

Project Due Date : 6:00am 15 October 2018

Demos will commence 15 October 2018.

Blackboard : log on to myecc
click on Blackboard link
from the course list select the cs132 section for which you are registered
click on Lessons, from there click on CS132 Project1
Under assignment submission, you will want to attach files, attach your zipped file mentioned above.