Programmer Manual
Class Board

1. Problem Description

A Board consists of 20 x 20 tiles inside of a border. Each tile is separated by white space. A turtle object is placed on the board. The user interacts with the turtle on the board in order to change the board, such as moving the turtle, or using the turtle to draw on the board.

2. Class Board

The Board itself is a pointer to an array of pointers, setting up a 20 x 20 two dimensional array.

Private data members:

SIZE	the size of the board
floor	a double pointer used to construct the board
groundChar	the character used for the ground
turtleChar	the character used for the turtle
topCornerLeft,	
topCornerRight,	
botCornerLeft,	
botCornerRight,	
horWall,	
vertWall	characters used to build the border
turtle	a pointer to a Turtle object

Private member functions:

checkCollision	checks to make sure the turtle stays in bounds
initBoard	initializes the board with the appropriate characters

Public member functions:

Board()	onstructor for a Board object
~Board()	destructor for a Board object
printBoard	prints the board out
moveTurtle	moves the turtle linearly according to its direction
jumpTurtle	changes the turtle's x and y coordinates
turnTurtle	changes the turtle's direction
resetBoard	returns the board and turtle to their default state
draw	draws on the board according to the turtle's movements
togglePen	moves the pen up or down
printTurtleInfo	prints out current information about the turtle

3. High Level Program Solution

Board()

Allocate memory for a 20 x 20 board
Initialize board with default values by calling initBoard()

initBoard()

Initializes all of the tiles of the board to the default ground character

```
printBoard()
    Prints out the top border
    Prints out the board
        If a tile has the Turtle on it, print out the turtle's character
        If a tile has not been drawn on, print out the ground character
        Otherwise, print out the tile as is
    Print out the bottom border

moveTurtle()
    IN:     numSpaces   the number of spaces to move the turtle

    Call the turtle's move function
    Call the checkCollision function to see if the turtle hits a wall
    If the turtle's pen is down, call the draw function to draw on the board

jumpTurtle()
    IN:     newX        the x position to jump to
            newY        the y position to jump to

    Call the turtle's jump function
    Call the checkCollision function to see if the turtle hits a wall
    If turtle's pen is down, draw on the tile landed on

turnTurtle()
    IN:     dir         the direction to turn the turtle

    Call the turtle's turn function

resetBoard()
    Set the turtle's position back to it's default position
    Set the turtle's direction back to it's default direction
    If the turtle's pen is down, toggle it back up
    Call the initBoard function to reinitialize the board

draw()
    Set the floor tile between the turtle's previous position and the turtle's current position to the draw character

togglePen()
    Call the turtle's changePenPosition function
    Change the current tile to the draw character

checkCollision()
    If the turtle's x position is outside the board, set the current x position to the maximum or minimum allowed position inside the board
    If the turtle's y position is outside the board, set the current y position to the maximum or minimum allowed position inside the board
```

```
changeBrush()
        IN:     selection       the number of the brush to be drawn with

        Set the draw character to the character returned by the turtle's changeBrush function

printTurtleInfo()
        Print out the current position of the turtle
        Get the current direction of the turtle, and print it out
        Get whether the turtle's pen is up or down and print it out
```