

cis1.5-spring2008-azhar, lab I, part 1

instructions

- This is the first part of the first homework/lab assignment for cis1.5.
- The entire assignment will be worth 9 points.
- The first part is worth 4 points and will be distributed and worked on in class on Mon February 4.
- The second part is worth 5 points and will be distributed and worked on in class on Thursday February 7.
- There will also be time to work on the assignment in class on Thursday February 14.
- **Both parts together are due on Thursday, February 14** and must be submitted by email (as below).
- **Follow these emailing instructions:**
 1. Create a mail message addressed to *mqazhar@sci.brooklyn.cuny.edu* with the subject line **cis1.5 hw1**.
 2. Attach **ONLY** the **.cpp** files for each part, as outlined below.
DO NOT ATTACH THE **.cbp** (CodeBlocks Project) files!
 3. Failure to follow these instructions will result in points being taken away from your grade. The number of points will be in proportion to the extent to which you did not follow instructions... (which can make it a lot harder for me to grade your work)

1 creating your first program in CodeBlocks

(2 points)

- Follow the “How to use CodeBlocks” instructions to create a new project called **roomba**.
- Type in the C++ code from the first roomba program example we created in class last week (on Thursday January 31). For your convenience the code is posted on the class web page (follow the “syllabus” link). It is also listed below in Appendix A.
- As instructed in the “How to use CodeBlocks” instructions, **Compile**, **Build** and try to run your program. Note that the “How to” document is designed for Mac OS X. If you are running on a Windows or Linux computer, you should be able to run the program by selecting **Build** and then **Run** from the CodeBlocks menus.
- Make sure the program runs as expected (i.e., see the example output on the last screen of the “How to” document).
- Now try changing some things in the **main()** method of the program.
 - Instead of starting roomba at location (0, 1), try starting at (10, 10).
 - Instead of going forward twice, backward once, left and then right, try another sequence of commands (“method calls”) for your imaginary roomba.
 - Try starting roomba at location (3, -1). What is the shortest sequence of commands you can think of to get it to end up at location (-1, 3)? Try modifying the code to do this, compile, build and run it. Keep editing the code, repeatedly compiling, building and running, until it works and you get the results you are looking for (i.e., in this case that the roomba starts at (3, -1) and ends up at (-1, 3)).
- Save this last version of the code (the one where the roomba starts at (3, -1) and ends up at (-1, 3)). This version of the project will be submitted as the first part of your first homework/lab assignment. Remember to submit only the **.cpp** file, which is inside the CodeBlocks project folder (NOT the .cbp file!).

2 creating your second program in CodeBlocks

(2 points)

- Create a new project. Type in the C++ code from the second roomba program example we created in class last week (on Thursday January 31). For your convenience the code is posted on the class web page (follow the “syllabus” link). It is also listed below in Appendix B.
- Make sure the program compiles, builds and runs as expected.
- Now try changing some things in the program.
 - Instead of using the letters **F, B, L, R** to tell the roomba to go Forward, Backward, Left and Right, modify the program so that the roomba is instructed to go North, South, East and West.
 - This means modifying the letters that the user enters (to **N, S, E, W**).
 - You should also modify the names of the methods so that your code is consistent. For example, instead of `moveForward()`, you should have `moveNorth()`.
- Save this version of the code (the one where the roomba responds to N, S, E and W). When it compiles, builds and runs properly, submit it. Remember to submit only the `.cpp` file, which is inside the CodeBlocks project folder (NOT the `.cpb` file!).

lab I, part 2

... will be distributed in class on Thursday February 7. You will also have time to finish this part on Thursday, February 14th.

Appendix A

```
// section 1: include C++ library definitions
#include <iostream>
using namespace std;

// section 2: declare variables
int x, y;

// section 3: declare methods
int display() {
    cout << "the roomba is at location (";
    cout << x;
    cout << ", ";
    cout << y;
    cout << ")\n";
    return 0;
}

int moveForward() {
    cout << "moving forward...\n";
    y = y + 1;
    return 0;
}

int moveBackward() {
    cout << "moving backward...\n";
    y = y - 1;
    return 0;
}

int moveLeft() {
    cout << "moving left...\n";
    x = x - 1;
    return 0;
}

int moveRight() {
    cout << "moving right...\n";
    x = x + 1;
    return 0;
}

// section 4: define main method
int main() {
    x = 0;
    y = 1;
    display();
    moveForward();
    moveForward();
    display();
    moveBackward();
    display();
    moveLeft();
    display();
    moveRight();
    display();
    return 0;
}
```

Appendix B

```
// section 1: include C++ library definitions
#include <iostream>
using namespace std;

// section 2: declare variables
int x; // robot's x position
int y; // robot's y position
char c; // user's input
bool q; // does user want to quit?

// section 3: declare methods
int display() {
    cout << "the roomba is at location (";
    cout << x;
    cout << ",";
    cout << y;
    cout << ")\n";
}

int moveForward() {
    cout << "moving forward...\n";
    y = y + 1;
}

int moveBackward() {
    cout << "moving backward...\n";
    y = y - 1;
}

int moveLeft() {
    cout << "moving left...\n";
    x = x - 1;
}

int moveRight() {
    cout << "moving right...\n";
    x = x + 1;
}

// section 4: define main method
int main() {
    x = 0;
    y = 0;
    q = false;
    while ( q==false ) {
        cout << "which way should roomba move (enter F,B,L,R or Q)? ";
        cin >> c;
        cout << "you entered: ";
        cout << c;
        cout << "\n";
        if ( c=='F' ) {
            moveForward();
            display();
        }
        else if ( c=='B' ) {
            moveBackward();
            display();
        }
    }
}
```

```
}
else if ( c=='L' ) {
    moveLeft();
    display();
}
else if ( c=='R' ) {
    moveRight();
    display();
}
else if ( c=='Q' ) {
    q = true;
}
else {
    cout << "oops! you entered something invalid. please try again :-)\n";
}
}
}
```