

CIS 1.5 Fall 2008 Lab I, Part 1

Instructions

- This is the first part of the first homework/lab assignment for CIS 1.5.
- The entire assignment will be worth 10 points.
- The first part is worth 5 points and will be distributed in class on Wednesday September 3rd.
- The second part is worth 5 points and will be distributed in class on Monday September 8th.
- **Both parts together are due on Wednesday September 17th** and must be submitted by email (as below).
- **Follow these emailing instructions:**
 1. Create a mail message addressed to *parsons@sci.brooklyn.cuny.edu* with the subject line **CIS 1.5 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 — grrrr!)

1 Creating your first program in CodeBlocks

(2 points)

- Follow the “How to use CodeBlocks” instructions to create a new project called **myhello**.
- Copy the C++ code from Appendix A into a file in the **myhello** project.
Make sure you change the comment to include your name.
- 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.
If you have trouble with doing this, please ask your professor for help.
- Does the program run as it did when we ran it in class?
- Change the messages that the program prints out. Compile the program, build and run it.
- Save this last version of the code (the one with the new messages). This version of the project will be submitted as the part of your first homework/lab assignment. Remember to submit only the **.cpp** file, which is inside the CodeBlocks project folder (NOT the .cpb file!).

2 Creating a second program in CodeBlocks

(1 point)

- Follow the “How to use CodeBlocks” instructions to create a new project called **mywords**.
- Write a C++ program in a file in the **mywords** project.

- Make sure you include a comment that gives your name, the date you wrote the program, the name of the file, and a short description of what the program does.
- The program should print 3 messages to the screen.
- There should be a blank line between each message.
- As instructed in the “How to use CodeBlocks” instructions, **Compile**, **Build** and run your program.
- Once your program works, save the final version. This version of the program will be submitted as part of your first homework/lab assignment. Remember to submit only the **.cpp** file, which is inside the CodeBlocks project folder (NOT the .cpb file!).

3 Creating a more complex program in CodeBlocks

(2 points)

- Follow the “How to use CodeBlocks” instructions to create a new project called **roomba**.
- Copy the C++ code from Appendix B into a file in the **roomba** project.
Your professor also has a copy of the code on a USB flash drive which you can use.
Make sure you change the comment at the start of the program to include your name.
- 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.
If you have trouble with doing this, please ask your professor for help.
- Does the program run as it did when your professor ran it?
- Now try changing some things in the **main()** method of the program.
 - Instead of starting the roomba at location (3, 4), try starting at (10, 10). Compile the program, build and run it.
 - Instead of making the roomba go north once, make it go north three times. Compile the program, build and run it.
 - Now make the roomba go north, then east twice, then south. Compile the program, build and run it.
 - 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 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!).

Lab I, part 2

... will be distributed in class on Monday September 8th.

Appendix A

```
//-----  
//  myhello.cpp  
//  
//  This program demonstrates output in C++  
//  
//  Simon Parsons  
//  2nd September 2008  
//-----  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
  
    cout << "Hello from my C++ world\n";  
    cout << "Here I write programs\n";  
  
    return 0;  
}
```

Appendix B

```
//-----  
//  
// roomba.cpp  
//  
// A small robot simulation program, which describes the  
// behavior of a simple roomba-like robot.  
//  
// Written by: Simon Parsons  
// 2nd September 2008  
//-----  
  
// C++ library definitions  
  
#include <iostream>  
using namespace std;  
  
// Variables  
  
int x; // The roomba's x co-ordinate  
int y; // The roomba's y co-ordinate;  
  
// Methods  
  
int displayPosition()  
{  
    cout << "Roomba is at: (";  
    cout << x << ", " << y;  
    cout << ")\n";  
}  
  
int goNorth()  
{  
    cout << "Moving north\n";  
    y = y + 1;  
}  
  
int goSouth()  
{  
    cout << "Moving south\n";  
    y = y - 1;  
}  
  
int goEast()  
{  
    cout << "Moving east\n";  
    x = x + 1;  
}  
  
int goWest()  
{  
    cout << "Moving west\n";  
    x = x - 1;  
}  
  
int main()  
{
```

```
cout << "Hello from your living room\n";

// initial position.

x = 4;
y = 3;

displayPosition();
goNorth();
displayPosition();

return 0;
}
```