

CIS 1.5 Fall 2009 Homework I

Instructions

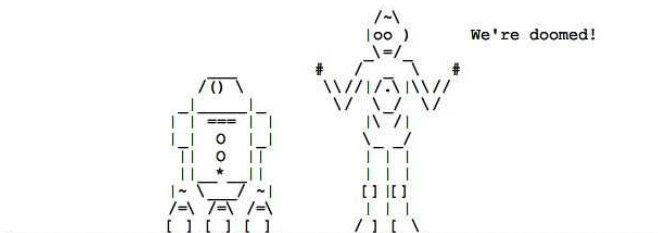
- This is the first homework/lab assignment for CIS 1.5.
- The entire assignment will be worth 5 points, that is 5% of your semester grade.
- **It is due by midnight on Thursday September 17th** and must be submitted by email (as below).
- **Follow these emailing instructions:**
 1. Create a mail message addressed to *parsons@sci.brooklyn.cuny.edu* with the subject line **CIS 1.5 HW1**.
 2. Write your name, that is the name under which you registered for the course, in the email (when I get an email from *deathmetal@aol.com* or *pinkprincess@yahoo.com*, I can usually guess whose program it is, but that is not as good as *knowing* whose program it is).
 3. Attach **ONLY** the **.cpp** files for each part, as I tell you below.
 4. Use a **zip** utility to bundle all your files together and send them as **ONE** attachment to the email.
 - on a PC: use **WinZip**
 - on a Mac: use **File - Create Archive...**
 - on Linux: use **zip**
 5. 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 Drawing with ASCII characters

(1 point)

- Have a look at the “ASCIIMATION” web site: <http://www.asciimation.co.nz/> (but don’t spend too long watching it...)

This site features “ASCII” animation, which is comprised of a series of many images that are cleverly drawn using only ASCII characters, like the image below:



- Create a temporary file (call it “tmp”) and design a picture using ASCII drawing.
- Then create a new C++ file, called **draw.cpp** and place inside the **main()** function, a series of **cout** statements that will produce the drawing you designed in your **tmp** file.

Hint: you will probably have to output blank spaces to get things to line up the way you want

- Compile and run your code, to make sure it does what you want. Edit again, as needed.

2 Weathering the storm

(1 point)

- Below is a 5-day weather forecast:

5-Day Forecast			Personalize your weather		
	Wednesday	Thursday	Friday	Saturday	Sunday
					
HI	31°F/-1°C	23°F/-5°C	36°F/2°C	50°F/10°C	54°F/12°C
Lo	14°F/-10°C	17°F/-8°C	29°F/-2°C	37°F/3°C	35°F/2°C
	Breezy and cold with clouds and sun	Mostly sunny, brisk and quite cold	Mostly sunny and not as cold	Partly sunny and milder	Mostly cloudy

- Write a program that computes (and outputs) the average high temperature and the average low temperature for those 5 days.
Hint: think about this on paper before you start typing on the computer. See if you can design the code in your notebook first, then type it in and see how it works. Note where you need to make corrections.
Hint: you will need to use a double variable for storing the averages.
- Name your program **weather.cpp**.

3 Living in a robot's world

(2 points)

- Imagine that you have a robot that lives in a 2-dimensional world that is made up of grid squares. The world is 12×12 squares in size, like this:

	1	2	3	4	5	6	7	8	9	10	11	12
1												
2												
3												
4												
5							R					
6												
7												
8												
9												
10												
11												
12												

The "R" shows where the robot is currently sitting in its world; in this case, it is in column 7 of row 5. We can express this as an (x, y) coordinate, where x is the column and y is the row; which means that, above, the robot is sitting in grid square $(7, 5)$.

- The robot can move in any of four directions: north, south, east or west. Each time it moves, it moves one square in the direction indicated by the command.

For example, if the robot, as above, is in grid square (7, 5) and it moves north, it will end up in grid square (7, 4).

- Write a program that computes the location of the robot, as follows. Assume that the robot's initial location is (7, 5).

- Define two integer variables called x and y which represent the robot's current location.
- Print out the robot's current location, in a friendly format. For example:

The robot is located at (7,5).

- The robot moves one square north. Update the values of x and/or y accordingly and print out the robot's new location, as well as the command it received. For example:

The robot received a command to move North. It is now located at (7,4).

- The robot moves one square east. Update the values of x and/or y accordingly and print out the robot's new location, as well as the command it received.
- The robot moves one square north. Update the values of x and/or y accordingly and print out the robot's new location, as well as the command it received.
- The robot moves one square west. Update the values of x and/or y accordingly and print out the robot's new location, as well as the command it received.
- The robot moves one square south. Update the values of x and/or y accordingly and print out the robot's new location, as well as the command it received.

- Name your program **robot1.cpp**

4 Submission

- You will be submitting three files:
 - draw.cpp
 - weather.cpp
 - robot1.cpp
- Make sure that you have a **comment** at the top of each file that contains the name of the file, **your name**, "CIS 1.5 HW 1" and the submission date (September 17th 2009).
- As above, use the zip utility to create a single archive containing all three files. Attach that archive to an email message to me.
- The subject line of your email should say: CIS 1.5 HW 1