# cis1.5-spring2007-sklar, lab VI

## instructions

- This is the LAST LAB of the semester! It covers the material in unit IV: simple classes.
- The assignment is worth 9 points and will be distributed and worked on in class on Wednesday May 9, Monday May 14 and Wednesday May 16.
- **The assignment is due on Monday May 21** and must be submitted by email (as below).
- **Follow these emailing instructions:**
  1. Create a mail message addressed to *sklar@sci.brooklyn.cuny.edu* with the subject line **cis1.5 hw6**.
  2. Attach ONLY the **.cpp** file, as outlined below.
     DO NOT ATTACH THE **.cbp** (CodeBlocks Project) file!
  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!)

## creating a simple class.

For this assignment, you will write a program that involves defining a simple class and creating an array of objects (i.e., variables) of your class.

Let's pretend that we are writing a program to control a sophisticated robot that will be used to help locate victims of a natural disaster. For example, you might be a rescue worker in Kansas helping to find victims buried in the rubble from the recent tornadoes that occurred there last week, and your robot will help you accomplish your task. Your robot has multiple sensors:

- temperature sensor—to measure body heat, which can help detect if a human is nearby
- motion sensor—to detect if something nearby is moving
- sound sensor—to "listen" for noises
- red color sensor—to indicate if anything *red* is within the robot's visual range

Your robot operates by wandering around the wreckage and every second, it's sensors take a measurement. Your job, for this assignment, is to create a data structure — a simple class — that will be able to store the measurements of your robot's sensors. Then you will create an array of these, so that, for example, you could store ten sets of measurements over a 10-second time period.

Note that you do not have to worry about the robot's motion (i.e., moving), you only need to worry about storing the robot's sensor readings.

Make sure you refer to the examples we discussed in class on Monday May 7!

Name your program: **sense.cpp** and submit ONLY this file.

*(continued on other side)*

1. Define a simple class called `sensors` that has the following data members:

   - `temperature`, which is a real number indicating the temperature near the robot;
   - `motion` which is a real number indicating the distance from the robot to the closest moving object (a positive value $\geq 0$); if there is nothing moving near the robot, then this value is set to the constant `NO_MOTION`$= -1$ (note that you will also need to define this constant);
   - `sound`, which is a positive whole number (integer) indicating the volume of any sounds being made near the robot; if there are no sounds, then this value is $0$;
     and
   - `red`, which is a boolean value indicating if there is anything the color red within the robot's visual range (i.e., true or false)

   *(2 points)*

2. Write a function:
   ```
   void setTemperature( sensors &s )
   ```
   that initializes the value of the `temperature` field in the argument `s` object to a random number between $32$ and $110$. *(1 point)*

3. Write a function:
   ```
   void setMotion( sensors &s )
   ```
   that initializes the value of the `motion` field in the argument `s` object to a random number between $-1$ (`NO_MOTION`) and $20$. *(1 point)*

4. Write a function:
   ```
   void setSound( sensors &s )
   ```
   that initializes the value of the `sound` field in the argument `s` object to a random number between $0$ and $10$. *(1 point)*

5. Write a function:
   ```
   void setRed( sensors &s )
   ```
   that initializes the value of the `red` field in the argument `s` object to either `true` or `false`, set randomly (hint: choose a random number between $0$ and $1$, and then use that to set the boolean value). *(1 point)*

6. Write a function:
   ```
   void readSensors( sensors &s )
   ```
   that calls the above functions (`setMotion()`, `setTemperature()`, `setSound()`, and `setRed()`) to set each of the fields in the argument sensor object `s`. *(1 point)*

7. Write a function:
   ```
   void writeSensors( sensors s )
   ```
   that displays on the screen the value of each of the fields in the argument sensor object `s`. *(1 point)*

8. Write the `main()` function that:

   - Declares an array of 10 `sensor` objects.
   - Calls `readSensors()` ten times, storing one set of sensor readings in each of the 10 array entries.
   - Then, AFTER all the reading of sensor values is done, calls `writeSensors()` to write on the screen all the sensor data that was collected in the previous step.

   *(1 point)*

Compile, build and run your program to make sure it works as you expect it to.