

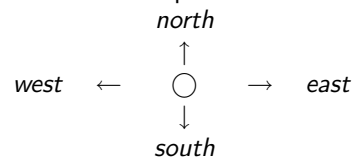
overview

- This is the project for unit IV of cis 1.5. This project covers the usage of arrays and strings.
- The project is worth 10% of your term grade. It will be marked out of **10 points**.
- The project is due via email on **Friday April 24**.
- Email your C++ file (**avoid.cpp**, as described below) to: sklar@sci.brooklyn.cuny.edu.

project description

This project simulates a robot wandering around and trying to avoid bumping into anything. This is a typical robot behavior called *obstacle avoidance*. The robot has 4 *distance* sensors that point in 4 directions:

Each sensor returns a value between 0 and 100. The smaller the value, the closer the robot is to an object. The bigger the value, the further the robot is from an object. In order to avoid obstacles, the robot should always move in the direction of the sensor with the largest value.



Below is part of an obstacle avoiding robot simulation program. Your job is to fill in the missing pieces.

```
#include <stdlib.h>
#include <sys/time.h>
#include <iostream>
#include <string>
using namespace std;

// function prototypes
int senseDistance();
void getSensors( int sensors[] );
void printSensors( int sensors[], string labels[] );
int furthest( int sensors[] );

// main function
int main() {
    int sensors[4];           // array of four distance sensors
    string labels[4];        // array of labels for each sensor
    int dir;                 // indicates which direction to go next
    srand( time( NULL ) );   // initialize the random number generator
    labels[0] = "north";     // initialize the sensor labels
    labels[1] = "west";
    labels[2] = "south";
    labels[3] = "east";
    for ( int i=0; i<5; i++ ) { // simulate 5 moves by the robot
        getSensors( sensors ); // get values for all the sensors
        printSensors( sensors, labels ); // print out the sensor values
        dir = furthest( sensors ); // find the direction that is furthest from an obstacle
        cout << "moving " << labels[dir] << endl; // 'go'
    }
} // end of main()
```

1. Create a file called **avoid.cpp** and type in the code above. You don't need to type in all the comments, unless you want to :-)

2. Complete the function `senseDistance()`, which should return a random integer between 0 and 100.
3. Complete the function `getSensors()`, which should assign a distance value to each element in the `sensors[]` array, by calling `senseDistance()` to get the distance value for each sensor.
4. Complete the function `printSensors()`, which should print out the value of each sensor, preceded by its label. For example, if `sensors[0] = 93`, `sensors[1] = 80`, `sensors[2] = 73` and `sensors[3] = 28`, then the output might look like this:

```
sensors = [north=93] [west=80] [south=73] [east=28]
```

Make sure that you use the `labels` array and don't hardcode any labels when printing out the sensor values (i.e., output "north" by using `labels[0]`).
5. Complete the function `furthest()`, which should return the *index* of the entry in the `sensors` array that has the largest value. For example, given the value of `sensors[]` shown above, the output from `furthest()` should be 0 because `sensors[0]` is 93, which is larger than the other elements in the `sensors[]` array.

submission instructions

- You will be submitting ONE file: **avoid.cpp**
- Make sure that you have a COMMENT at the top of the file that contains the name of the file, YOUR NAME, "CIS 1.5 PROJECT 4" and the submission date (April 24, 2009).
- The SUBJECT LINE of your email should say: CIS 1.5 PROJECT 4
- The BODY of your email should contain your name.