

## 1 Introduction

For this lab you will be writing a simple robot class. Your robots will understand basic commands like `turnLeft` and `goForwards`. This handout will specify all of the functionality that the robots should have, and it will give you a sample `main` for you to check your work.

This lab is also your **last CSI 201 assignment!** It is due on the last day of classes: **11:59pm, Thursday December 8th**. As usual, upload your `.cpp` file to canvas.

Aside from a `main`, we are not providing any code. Almost all of your work should go inside `class Robot`, which you will be writing.

## 2 Robot Class Specifications

Your `Robot` class should keep track of the following variables. Each of them **must** be `private`. Use `doubles`.

- `xpos` - The current  $x$ -coordinate of where the robot is
- `ypos` - The current  $y$ -coordinate of where the robot is
- `odometer` - How far the robot has traveled since it powered on
- `heading` - An angle, in degrees, saying which direction the robot is pointing. We will think of 0 degrees as being pointing to the right, and 90 degrees as pointing up. (If you use the equations in this handout, you probably won't have to worry about this.)

Additionally, your `Robot` class should have all of the following `public` functions inside of it:

- `getOdometer()` - returns the current odometer reading
- `powerOn()` - sets all four private variables to 0.0
- `turnLeft(double t)` - increases `heading` by `t`
- `turnRight(double t)` - decreases `heading` by `t`
- `goForwards(double d)` - moves the robot forwards, which involves:
  - increasing `xpos` by `d * cos(heading)`
  - increasing `ypos` by `d * sin(heading)`
  - increasing `odometer` by `d`
  - Watch out! The `sin` and `cos` functions in `cmath` expect inputs in radians, and our heading numbers are all in degrees. You will have to figure out (i.e., google) how to do a degrees-to-radians conversion, or else your `goForwards` won't work right at all.
- `printStatus()` - print out where the robot is, and what its heading is.

## 3 How to Know if it Works

Here is a sample `main`:

```
int main() {
    Robot x86;
    x86.powerOn();
    cout << "Checkpoint 1:" << endl;
```

```
x86.printStatus();
cout << endl;

x86.turnLeft(132);
x86.goForwards(23);

cout << "Checkpoint 2:" << endl;
x86.printStatus();
cout << endl;

x86.turnRight(95);
x86.goForwards(35);
x86.turnRight(103);
x86.goForwards(8);
x86.turnRight(16);
x86.goForwards(219);

cout << "Checkpoint 3:" << endl;
x86.printStatus();
cout << endl;

cout << "The robot odometer reading is " << x86.getOdometer() << endl;
}
```

The correct output from this main is:

```
Checkpoint 1:
Robot at (0,0) with heading 0
Checkpoint 2:
Robot at (-15.39,17.0923) with heading 132
Checkpoint 3:
Robot at (46.295,-186.021) with heading -82
The robot odometer reading is 285
```