

CS16, 10S, **H14**, due **Fri Lecture 04.30**—More on structs (handout)—Total Points: 50

Available online as <http://www.cs.ucsb.edu/~pconrad/cs16/10S/homework/H14>—printable [PDF](#)

Name: (4 pts)	Umail Address: (4 pts)	@umail.ucsb.edu
_____	_____	

Lab Section (2 pts)—circle one: 9am 10am 11am noon unknown

(Note: For now, circle the lab section you are registered for on GOLD. If you need to request attendance at a different lab section because of an ACTUAL SCHEDULE CONFLICT, please email pconrad@cs.ucsb.edu with details)

This assignment is due **IN Lecture on Friday, 04.30**.
It may ONLY be submitted Lecture, in Chem 1171 at 1pm on Friday.
You must come IN PERSON to turn it in during your assigned Lecture section.

Late Policy: No email submission allowed—and don't "slip it under my door". If you need to make it up, you must do so during office hours, or make an appointment to see me, and you must request this appointment within 48 hours of when the assignment was originally due.

Personal Day/Sick Day policy: Everyone is permitted one "personal day/sick day" when you get to make up a missed homework assignment for free during office hours or via appointment. After that, you may not make up the homework assignment—you can only earn back the points through extra credit opportunities.

(For more details, see the [syllabus](#) and the [homework policy](#))

For this homework, the preparation is material on the following handout:
<http://www.cs.ucsb.edu/~pconrad/cs16/10S/homework/H14/handout> ([pdf](#) link)

**Be sure to read not only the main text on the handout,
but also the little boxes off to the side, like the one see to the right of this sentence.**

Once you've read that handout, write answers to the questions on this sheet
(use the [PDF link](#) to print a copy of this if you weren't in class).

1. (10 pts) Assuming a `struct Time` is defined like this:

```
struct Time
{
    int hrs;
    int min;
};
```

Write a function definition for a function with this prototype:

```
void printTime(struct Time t);
```

that for a struct where the value of `hrs` is, for example, 9, and the value of `min` is 20, prints:

```
09:20
```

and for a struct where the value of `hrs` is, for example, 17 and the value of `min` is 7, prints:

```
17:07
```

A `printf` tip that may help: you can use `printf("%02d", x);` to print an `int` expression `x` in a field of exactly two characters, with a single zero in front of single digit positive integers, and `"00"` for the number 0.

The little boxes on the [handout](#)—boxes like this one—also have important information you may need to complete the assignment.

Please turn over for more...

...continued from other side

2. (10 pts) Write a struct definition for a struct `Polar` that can hold the representation of a point in polar coordinates.

As a reminder, in polar coordinates, a point is represented by two values, r and θ (theta), as illustrated in the pictures shown at right:

- r is the distance of the point from the origin (or, the radius of the circle centered at the origin on which the point lies)
- θ is the angle between the x-axis and a line from the origin to the point

The members of the struct `Polar` should be called `r` and `theta` and should both be of type `double`.

3. (20 pts) Given the definition of struct `Polar` from the previous problem. along with this definition:

```
struct Point
{
    double x;
    double y;
};
```

Write the definition of a function called `PointToPolar` that takes a struct `Point` as its one argument, and returns a struct `Polar` as the result.

As a reminder, the formulas to convert from XY to Polar Coordinates are as follows:

$$r = \sqrt{y^2 + x^2}$$

$$\theta = \begin{cases} 0 & \text{if } x = 0 \text{ and } y = 0 \\ \arcsin\left(\frac{y}{r}\right) & \text{if } x \geq 0 \\ -\arcsin\left(\frac{y}{r}\right) + \pi & \text{if } x < 0 \end{cases}$$

You may assume that `#include <math.h>` is available, and thus you may use:

- the function with prototype `double sqrt(double x);` to compute square root
- the pre-defined constant `M_PI` for the value of π
- the function with this prototype to compute arcsin:
`double asin(double x);` // x must be in the range -1 to 1

