# Arduino Programming
# Part 6:
# LCD Panel Output

ME 121

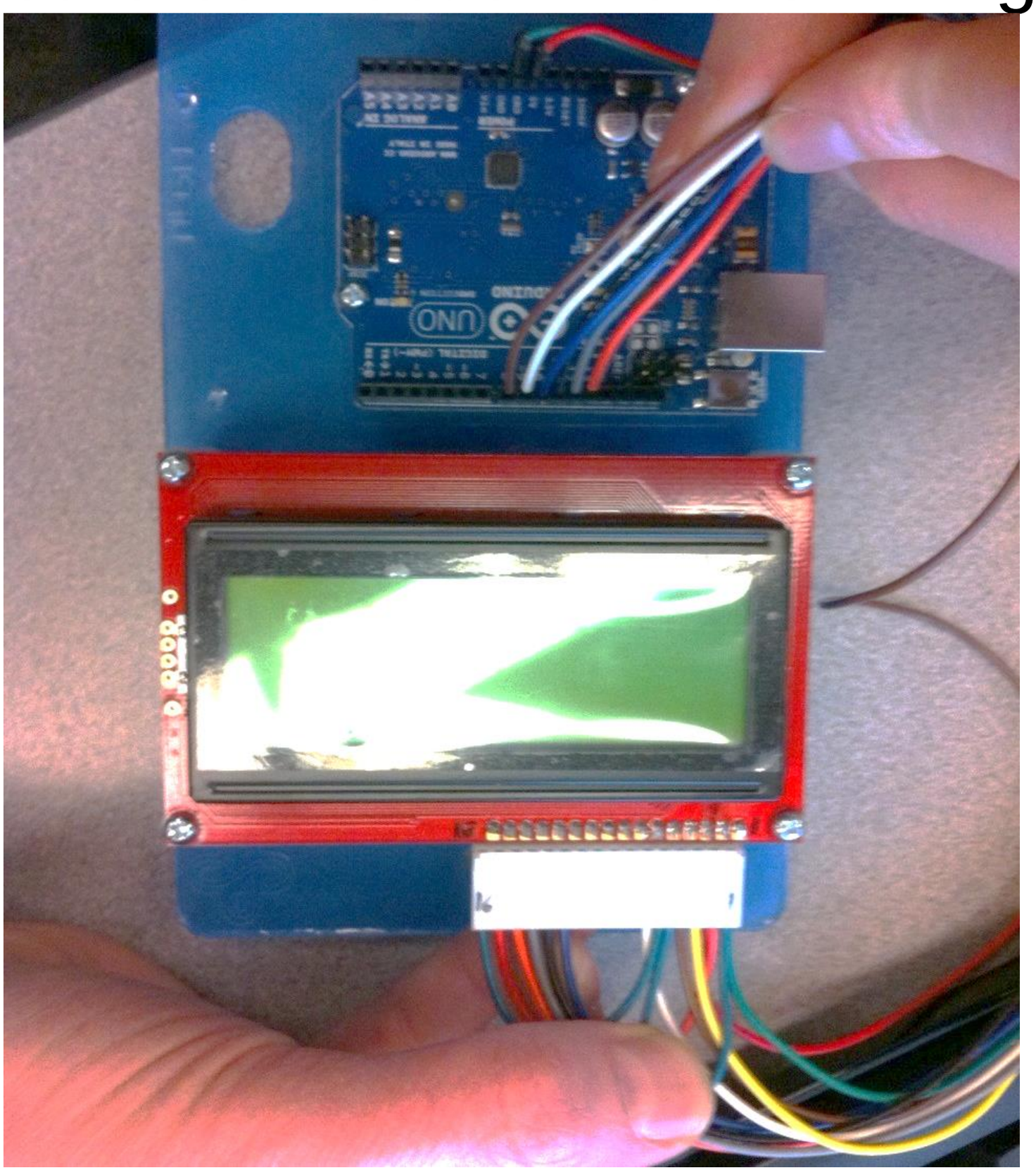
Portland State University

# Goals

Use the 20x4 character LCD display for output

- Overview of assembly — detailed instructions on the web
  - http://web.cecs.pdx.edu/~me121/doku.php?id=lecture:wiring_harness
  - http://www.ladyada.net/learn/lcd/charlcd.html
- Introduction to the LCD library
  - http://www.arduino.cc/en/Tutorial/LiquidCrystal
- Simple demonstration
- Map the 20x4 character display for fish tank data

# Breadboard connection via the wiring harness



# Programming Arduino for LCD Display

Refer to Adafruit tutorial

- http://www.ladyada.net/learn/lcd/charlcd.html

and Arduino documentation

- http://www.arduino.cc/en/Tutorial/LiquidCrystal

# Test the display

File ⟹ Examples ⟹ LiquidCrystal ⟹ HelloWorld

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
      // set up the LCD's number of columns and rows:
      lcd.begin(16, 2);
      // Print a message to the LCD.
      lcd.print("hello, world!");
}

void loop() {
      // set the cursor to column 0, line 1
      // Line 1 is the second row, because counting begins with 0
      lcd.setCursor(0, 1);
      // print the number of seconds since reset:
      lcd.print(millis()/1000);
}
```

# Test the display

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
      // set up the LCD's number of columns and rows:
      lcd.begin(16, 2);
      // Print a message to the LCD.
      lcd.print("hello, world!");
}

void loop() {
      // set the cursor to column 0, line 1
      // Line 1 is the second row, because counting begins with 0
      lcd.setCursor(0, 1);
      // print the number of seconds since reset:
      lcd.print(millis()/1000);
}
```

Change pin assignments to match wiring harness: (8,9,10,11,12,13)

Change to (20,4)

File ⟹ Examples ⟹ LiquidCrystal ⟹ HelloWorld

# Test the display

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
      // set up the LCD's number of columns and rows:
      lcd.begin(16, 2);
      // Print a message to the LCD.
      lcd.print("hello, world!");
}

void loop() {
      // set the cursor to column 0, line 1
      // Line 1 is the second row, because counting begins with 0
      lcd.setCursor(0, 1);
      // print the number of seconds since reset:
      lcd.print(millis()/1000);
}
```

lcd is a LiquidCrystal object

File ⟹ Examples ⟹ LiquidCrystal ⟹ HelloWorld

# Arduino code to write to the LCD panel

## Include the LCD library

In the header:
(outside and before setup)

```
#include <LiquidCrystal.h>
```

## Initialize the display by creating a LiquidCrystal object

Before using the display:

```
LiquidCrystal lcd(p1,p2,p3,p4,p5,p6);
lcd.begin(20,4);
```

## Send characters in a two-step process

Move the cursor:

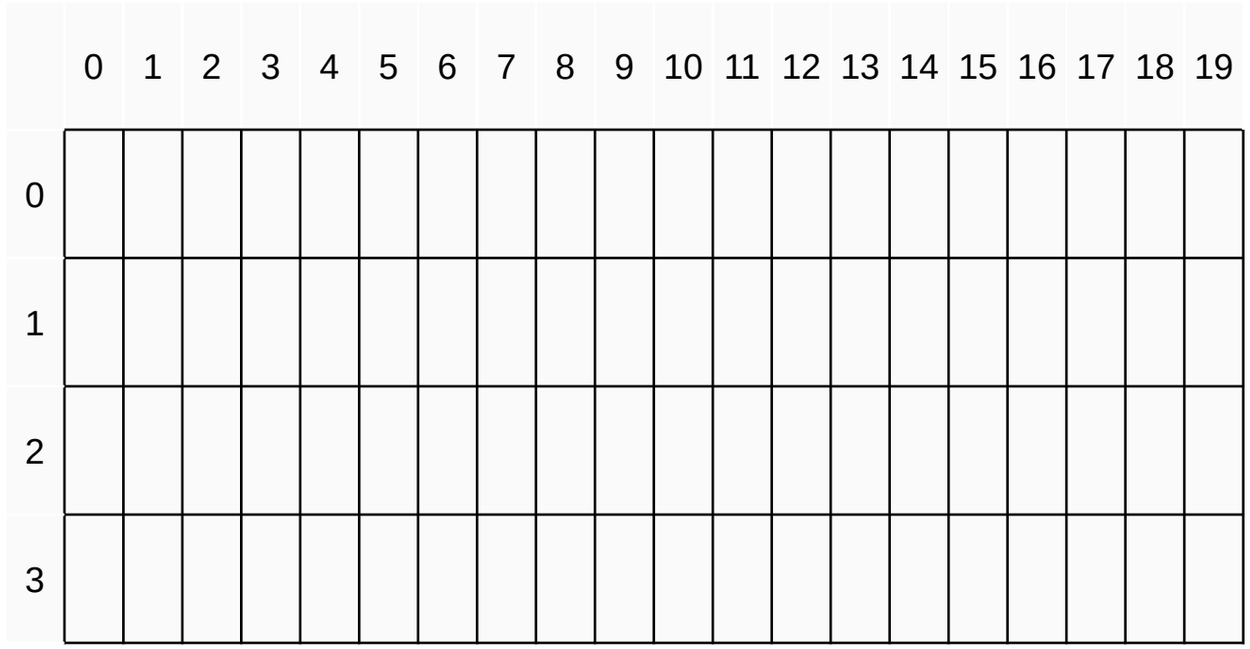
```
lcd.setCursor(column,row)
```

Display the message:

```
lcd. print("message")
```

# Character matrix on a 4 X 20 display

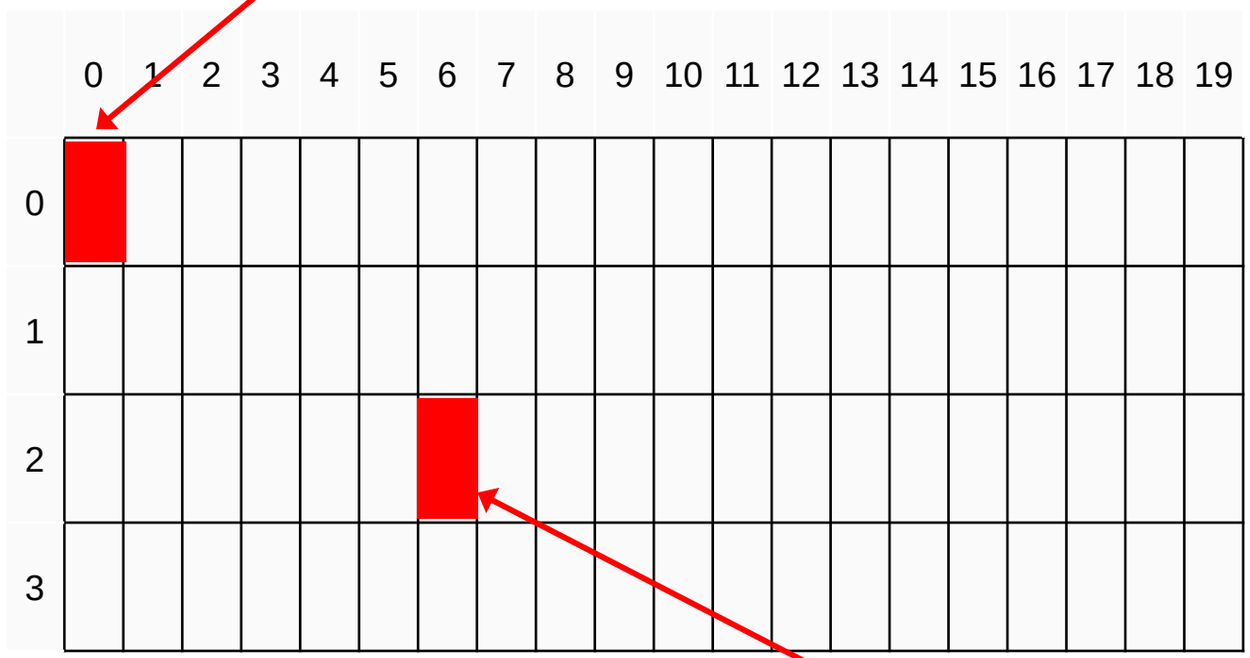
Row and column indices begin with zero



# Character matrix on a 4 X 20 display

Row and column indices begin with zero

lcd.setCursor (0,0)




lcd.setCursor (6,2)

# Display fish tank salinity

Modify the HelloWorld code to display the salinity

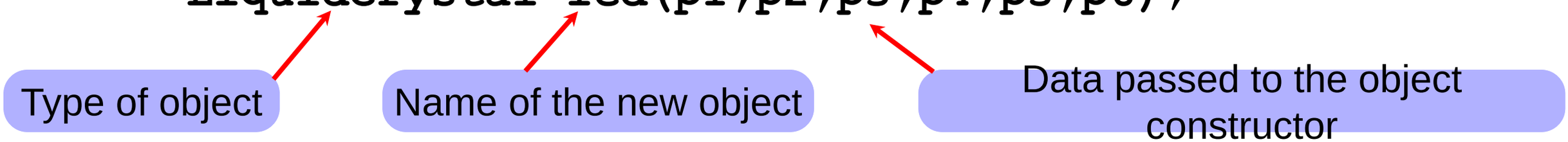
- “Salinity = ” and “Average of ” can be displayed once at the start
- x.xx and NNN values change, and are updated on the display.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | S | a | l | i | n | i | t | y | | = | x | . | x | x | % | | | | | |
| 1 | A | v | e | r | a | g | e | | o | f | | N | N | N | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | |

# LCD library code

Create a new LiquidCrystal object:

```
LiquidCrystal lcd(p1,p2,p3,p4,p5,p6);
```




When a new object is created, the data passed to the constructor is *stored in* the object. Thus, whenever we use the variable `lcd` again in the program, the `lcd` object “knows” that it is connected to `p1, p2, ..., p6`.

# LCD library code

Tell the lcd object about the size of the display

```
lcd.begin(20,4)
```

Run the “begin” method

Pass the values 20 and 4 to the “begin” method

Objects have data and methods

- Data are values associated with a particular “instance” of an object
- Some data may be “public”. Programmers can view or change public data.
- Some data may be “private”, and therefore unavailable to programmers.
- Methods are functions that an object knows how to perform
  - Methods can return values
  - Methods can change public data
  - Methods can perform computations and interact with the environment (sensors)

# LCD library code

Change the current cursor position:

```
lcd.setCursor(12,1)
```

Run the “setCursor” method

Pass 12 and 1 to the “setCursor” method

The setCursor methods prepares lcd for its next action

```
lcd.print(“Hello”)
```

Run the “print” method

Use “Hello” as data for the print method

lcd.print(...) works because the lcd object “knows” about its current position (from setCursor), the size of the display (from begin), and from the pin assignments from the constructor. When the lcd.print() method runs, it unleashes action that is constrained by data stored in the object.