Outline

- What is C?
- Programming Target: Arduino
- Programming Basics
  - Simple C Program
  - Running a Program
- Programming Tools
  - Comments
  - Data Types
  - Variables
  - Console Inputs and Outputs
- More stuff you can do…
What is C?

- Created by Dennis Ritchie at Bell Laboratories in 1972
- Programming language for making a computer/microcontroller do something.
- One of the most popular programming languages:
  - Available for many platforms (supercomputers to embedded microcontrollers)
  - Relatively easy to use, moderate level of abstraction, but programmer also has an idea of how code will be executed
  - Can interact with hardware directly
Programming Target: Arduino

- Arduino
  - type of microcontroller
  - we’ll talk about this a lot more next time
- Overall syntax is same as C
- Some differences (we’ll highlight them)
- FYI, Arduino’s version of C is called “Arduino”
void setup()
{
}

void loop()
{
}

This runs first
(set things up)

Then this runs repeatedly
(it loops)

All programs MUST contain these two functions
Simple C Program: Example

```c
void setup()
{
    Serial.begin(9600);
    Serial.println("Hello world!");
}

void loop()
{
}
```
Simple C Program: Example

```c
void setup()
{
  Serial.begin(9600);
}

void loop()
{
}
```

Setup serial port to run at 9600 baud (bits/second)

Print “hello world!” to the serial port (followed by a carriage return)

In this program, the `loop()` function does nothing (but still must be included!)
Running a Program on the Arduino

- Run the Arduino software: `arduino.exe`
  - [arduino.cc/en/Main/Software](https://arduino.cc/en/Main/Software)
- Type the program into the *sketch*
- Save the file using a meaningful name – e.g. “helloworld”
  - From the file menu: File -> Save As
  - The file will save with the .ino extension (helloworld.ino)
- Connect the Arduino board using an FTDI USB cable
  - Black wire goes to GND
- Change the settings to the correct device and port
  - From the file menu: Tools -> Board -> Arduino Uno
  - Check the USB port settings with Tools -> Serial Port
    - Random COM port on Windows
    - `/dev/tty.usbserial-randomcharacters` on mac
- Verify the code
- Upload the code
- Open the Serial Monitor (after the code uploads)
Running a Program on the Arduino (cont.)

- Run the Arduino software: `arduino.exe`
- Type the program into the `sketch`
Demo
Coding: Your Turn!

Write a program that repeatedly prints the phrase: “I love E11 already!”
Coding: Your Turn!

Write a program that repeatedly prints the phrase: “I love E11 already!”

```cpp
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    Serial.println(“I love E11 already!”);
}
```
Outline

- What is C?
- Programming Target: Arduino
- Programming Basics
  - Simple C Program
  - Running a Program
- Programming Tools
  - Comments
  - Data Types
  - Variables
  - Console Inputs and Outputs
- More stuff you can do...
Comments

- Are ignored by the computer running the program
- But are **critical** for clarity and organization
- Single-line comment
  ```
  // single-line comment
  ```
- Multiple-line comments
  ```
  /* multiple-line comment */
  ```
Data Types

• A data type tells us:
  – The **type** of values represented
  – The **range** of values
## Data Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Size (bits)</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>8</td>
<td>$-2^7$ (-128)</td>
<td>$2^7 - 1$ (127)</td>
</tr>
<tr>
<td>unsigned char</td>
<td>8</td>
<td>0</td>
<td>$2^8 - 1$ (255)</td>
</tr>
<tr>
<td>int</td>
<td>16</td>
<td>$-2^{15}$ (-32,767)</td>
<td>$2^{15} - 1$ (32,768)</td>
</tr>
<tr>
<td>unsigned int</td>
<td>16</td>
<td>0</td>
<td>$2^{16} - 1$ (65,535)</td>
</tr>
<tr>
<td>long</td>
<td>32</td>
<td>$-2^{31}$ (-2,147,483,648)</td>
<td>$2^{31} - 1$ (2,147,483,647)</td>
</tr>
<tr>
<td>unsigned long</td>
<td>32</td>
<td>0</td>
<td>$2^{32} - 1$ (4,294,967,295)</td>
</tr>
<tr>
<td>float</td>
<td>32</td>
<td>$\pm 2^{-126}$</td>
<td>$\pm 2^{128} \times (2-2^{-15})$</td>
</tr>
<tr>
<td>boolean</td>
<td>8</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

**Note:**
- byte = unsigned char
- double = float
- word = unsigned int
Binary Numbers: Range

• What happens when a result won’t fit in that range?
  – Overflow!
  – For example, with only 2 bits:
    \[ 11 + 01 = 100 = 00! \]
void setup()
{
    char x = 33;
    char y = 257;
    Serial.begin(9600);

    Serial.print("The value of x is ");
    Serial.println(x, DEC);

    Serial.print("The value of y is ");
    Serial.println(y, DEC);
}

void loop()
{
}
Variables

• Each variable has a:
  – Name
  – Type
  – Value

Example:

```c
int cnt = 5;  // name is cnt, type is int, value is 5
```
int count = 0;   // global variable

void setup() {
    char x;       // local variables
    float y = 7.8;
    boolean found = false;

    x = 12;      /* x is initialized after it is declared. */
    ...
}

Variables
Variables

• All variables must be *initialized* (set to a known value) before they are used
• Global variables:
  – are declared outside of all functions
  – are accessible anywhere in the program
• Local variables
  – are declared within a function
  – are only accessible within that function
Variables

```java
int cnt = 0;

void setup() {
    char x;
    float y = 7.8;
    boolean found = false;

    x = 12;
    ...
}

void loop() {
    cnt = 42;
    x = 3;
}
```
Coding: Your Turn!

Write a program that converts the variable \( x \) from centimeters to inches and prints the value of \( x \) in both units.

```cpp
// convert x from cm to in
int x = 12;
```
Coding: Your Turn!

Write a program that converts the variable x from centimeters to inches and prints the value of x in both units.

```cpp
// convert x from cm to in
int x = 12;
float y = 12/2.54;

Serial.print("x = "); Serial.print(x);
Serial.print(" cm = "); Serial.print(y);
Serial.println(" inches.");
```
Console Input and Output

- **Output**
  - `Serial.print(string or variable name);`
  - `Serial.println(string or variable name);`

- **Input**
  - `int Serial.read();`
```cpp
void setup() {
  Serial.begin(9600); // opens serial port at 9600 baud
  Serial.println("Enter a value: ");
}

void loop() {
  int incomingByte = 0; // incoming serial data

  // read user input
  if (Serial.available() > 0) {
    incomingByte = Serial.read();

    // print result:
    Serial.print("I received: ");
    Serial.println(incomingByte, HEX);
  }
}
```
In my eyes, that should be spelled ASCII

ASCII
```cpp
int incomingByte = 0; // incoming serial data

void setup()
{
    Serial.begin(9600); // opens serial port at 9600 baud
    Serial.println("Enter a value: ");
}

void loop() {
    // read user input
    if (Serial.available() > 0) {
        incomingByte = Serial.read();

        // print result:
        Serial.print("I received: ");
        Serial.println(incomingByte, DEC);
    }
}
```
• Print formats

Serial.print(val, format)
Serial.println(val, format)

– val is value to print (any data type)
– format is:
  – DEC (decimal)
  – HEX (hexadecimal)
  – OCT (octal)
  – BIN (binary)
  – BYTE (ASCII-interpreted byte)
  – or number of decimal places (for floating point)
Physical Inputs and Outputs

• **Setup:**
  – `pinMode(pin, mode)`
  – `mode` is either: INPUT or OUTPUT

• **Output – setting a pin value:**
  – `digitalWrite(pin, value)`
  – `value` is either: HIGH or LOW

• **Input:**
  – `digitalRead(pin)`
void setup()
{
  Serial.begin(9600);

  // set LED pin as output
  pinMode(13, OUTPUT); // LED pin
}

void loop()
{
  Serial.println("Testing LED");

digitalWrite(13, HIGH); // turn the LED on
delay(200);            // delay 200 ms

digitalWrite(13, LOW);  // turn the LED off
delay(200);            // delay 200 ms
}
void setup()
{
    Serial.begin(9600);

    // set speaker pin and LED as outputs
    pinMode(4, OUTPUT);  // speaker pin
    pinMode(13, OUTPUT); // LED pin
}
void loop()
{
    Serial.println("Testing speaker");

    tone(4, 440);           // write tone of 440 Hz to speaker
    digitalWrite(13, HIGH); // turn the LED on
    delay(200);             // delay 200 ms
    noTone(4);              // turn the speaker (pin 4) off
    digitalWrite(13, LOW);  // turn the LED off
    delay(200);             // delay 200 ms
}
Useful Resource!!!