Name: _____ Date: _____ Class: _____

Common Functions for Arduinos Reference Sheet

Function	Description	Example

setup()	Must be in any Arduino program for it to work. Anything contained in this function will be executed once.	void setup() {
		Serial.println("Hello!");
		}

loop()	Must be in any Arduino program for it to work. Anything contained in this function will be executed over and over again (infinitely).	void loop() {
		Serial.println("Row row row your boat!");
		}

Serial.begin()	To use the Serial Monitor, this function must be used to setup the Serial connection that it will use to communicate with the computer. The number inside the parentheses specifies the transmission rate; 9600 bps is most common.	Serial.begin(9600);
Serial.print() **&** **Serial.println()**	Can be used to print strings or variable values to the Serial monitor. println adds a carriage return at the end of what is printed.	Serial.print("Hello!"); Serial.println("I go to school."); Serial.print("I enjoy working with "); Serial.print("Arduinos."); /* This is what is printed: Hello! I go to school. I enjoy working with Arduinos. */

Serial.available()	Checks to see if information (data) is waiting in the Serial port buffer. It returns the number of bytes that are available to be read.	int a;
		a = Serial.available();
		//The variable "a" will be assigned the number of bytes of information waiting in the Serial port buffer.

Serial.read()	Reads the first byte of information (or first character) waiting in the Serial port buffer and assigns it to a variable. This function would have to be called repeatedly if more than one character is waiting in the buffer.	//Pretend that the user sent the letter "Q"
		char userInput;
		userInput = Serial.read();
		//userInput would now hold the character "Q"

pinMode()	Used to set digital I/O pins as either input or output. This is often done in the setup() function.	int ledPin = 5; int sensorPin = 6;

	Analog pins are automatically set to be input pins, so no need exists to use this function with them.	pinMode(ledPin, OUTPUT);
		pinMode(sensorPin, INPUT);
		
		//Pin 5 is now set to output signals (create voltage). Pin 6 is now set to receive signals (measure voltage).

digitalWrite()	This function is used to turn a digital I/O pin on (HIGH) or off (LOW).	int ledPin = 5;
		pinMode(ledPin, OUTPUT);
		digitalWrite(ledPin, HIGH);
		digitalWrite(ledPin, LOW);
		
		//Pin 5 is turned on and off. If an LED is connected to this pin, you would see it light up and then go out.

digitalRead()	Used to read a signal (measure a voltage) at a digital I/O pin. It only reads either HIGH (on) or LOW (off).	int sensorPin = 5;
		int val = 0;
		pinMode(sensorPin, INPUT);
		val = digitalRead(sensorPin);
		
		//If the sensor connected to pin 5 is on, val will receive "HIGH." If the sensor is off, val will receive "LOW."

analogRead()	This function is used to read a signal (measure a voltage) at an analog pin. It returns a value between 0 (off) and 1023 (on or high).	int sensorPin = A0;
		int val = 0;
		val = analogRead(sensorPin);
		
		//Whatever the voltage is at the analog pin 0, that value (represented as a number between 0 and 1023) will be stored in the variable val.

analogWrite()	This function is **NOT** used with analog pins; it is used with digital pins.
	
	Allows a digital pin to output a simulated voltage between 0V and 5V using pulse-width modulation (PWM). This can only be used on certain digital pins on the Arduino board (they have a ~ sign next to them).
	
	A value between 0 (always off, 0V) and 255 (always on, 5V) must be specified.
		int ledPin = 5;
		pinMode(ledPin, OUTPUT);
		
		analogWrite(ledPin, 127);
		
		//A voltage of approximately 2.5V is simulated and used to power whatever is connected to Pin 5.