ME 121 – Arduino Practice problems

Objective: You should be able to answer by hand (with no computer or cheat sheet) these problems.

Note: To check your result, copy/paste your code and check if you get the answer expected!

ARDUINO CHEAT SHEET

For more information visit: http://arduino.cc/en/Reference/

Structure

```
/* Each Arduino sketch must contain the
following two functions. */
void setup()
{
/* this code runs once at the beginning of
the code execution. */
}

void loop()
{
/* this code runs repeatedly over and over
as long as the board is powered. */
}
```

Comments

```
// this is a single line
/* this is
a multiline */
```

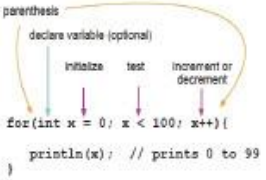
Setup

```
pinMode(pin, [INPUT \ OUTPUT \ INPUT_PULLUP]);
/* Sets the mode of the digital I/O pin.
It can be set as an input, output, or an
input with an internal pull-up resistor.
*/
```

Control Structures

```
if(condition)
{
// if condition is TRUE, do something here
}
else
{
// otherwise, do this
}

for(initialization; condition; increment)
{
// do this
}
/* The 'for' statement is used to repeat
a block of statements enclosed in curly
braces. An increment counter is usually
used to increment and terminate the loop.
*/
```



Digital I/O

```
digitalWrite(pin, val);
/* val = HIGH or LOW write a HIGH or a LOW
value to a digital pin. */
int var = digitalRead(pin);
/* Reads the value from a specified digital
pin, either HIGH or LOW. */
```

Analog I/O

```
analogWrite(pin, val);
/* Writes an analog value to a pin.
val = integer value from 0 to 255 */
int var = analogRead(pin);
/* Reads the value from the specified
analog pin. */
```

Advanced I/O

```
tone(pin, freq);
/* Generates a square wave of the specified
frequency to a pin. Pin must be one of the
PWM (~) pins. */
tone(pin, freq, duration);
/* Generates a square wave of the specified
frequency to a pin for a duration in
milliseconds. Pin must be one of the PWM (~)
pins. */
noTone(pin);
// Turns off the tone on the pin.
```

Time

```
delay(time_ms);
/* Pauses the program for the amount of time
(in milliseconds). */
delayMicroseconds(time_us);
/* Pauses the program for the amount of time
(in microseconds). */
millis();
/* Returns the number of milliseconds since
the board began running the current program.
max: 4,294,967,295 */
micros();
/* Returns the number of microseconds since
the board began running the current program.
max: 4,294,967,295 */
```

Data Types

```
void    // nothing is returned
boolean // 0, 1, false, true
char    // 8 bits: ASCII character
byte    // 8 bits: 0 to 255, unsigned
int     // 16 bits: 32,768 to 32,767, signed
long    /* 32 bits: 2,147,483,648
           to 2,147,483,647, signed */
float   // 32 bits, signed decimal
```

Constants

HIGH \ LOW
INPUT \ OUTPUT
true \ false

Mathematical Operators

```
=  // assignment
+  // addition
-  // subtraction
*  // multiplication
/  // division
%  // modulus
```

Logical Operators

```
==  // boolean equal to
!=  // not equal to
<   // less than
>   // greater than
<=  // less than or equal to
>=  // greater than or equal to
&&  // Boolean AND
||  // Boolean OR
!   // Boolean NOT
```

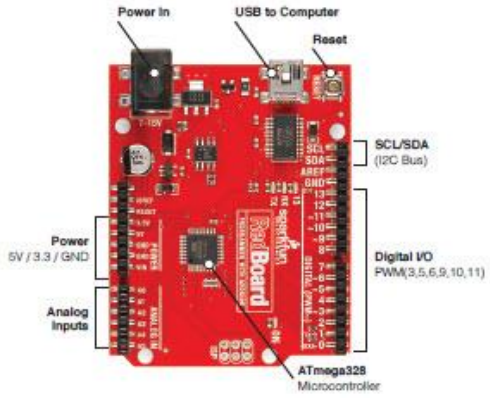
Bitwise Operators

```
&       // bitwise AND
|       // bitwise OR
^       // bitwise XOR
~       // bitwise INVERT
var << n   // bitwise shift left by n bits
var >> n   // bitwise shift right by n bits
```

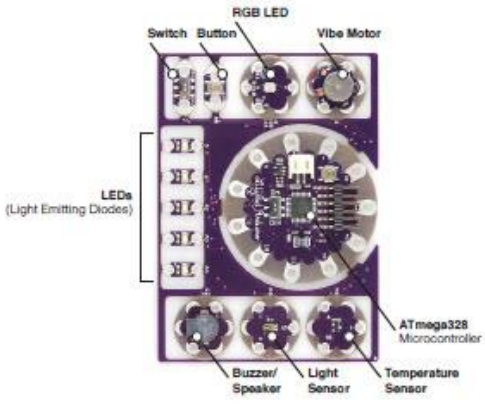
Libraries

```
#include <libraryname.h>
/* this provides access to special
additional functions for things such as
servo motors, SD card, wifi, or bluetooth.
*/
```

RedBoard:



LilyPad ProtoSnap Simple:



1. Variables: declaration and use

What would be the final values of x, y, z in the codes below:

```
void setup(){
 int x,y,z;
 x = 15;
 y = 10;
 z = x + y ;
}

void loop(){
}
```

```
void setup(){
 int x,y,z;
 x = 15.0;
 y = 10.0;
 z = x + y ;
}

void loop(){
}
```

```
void setup(){
 int x,y,z;
 x = 15.0;
 y = 10.0;
}

void loop(){
 z = x + y ;
}
```

```
void setup(){
 int x,y,z;
 x = 15;
 y = 10;
 z = x / y ;
}

void loop(){
}
```

```
void setup(){
 float x,y,z;
 x = 15;
 y = 10;
 z = x / y ;
}

void loop(){
}
```

```
void setup(){
 int x,y ;
 float z;
 x = 15;
 y = 10;
 z = x / y ;
}
void loop(){
}
```

2. Analog I/O

- Analog input: Draw a circuit to use a photoresistor. Then write a code that will read the values of a photoresistor and print them on the Serial Monitor continuously.
- Analog output: Draw a circuit to use an LED. Then write a code that will turn the LED on with high intensity for 1 second, then half intensity for 0.5 second, then off for 0.25 second, and repeat indefinitely.

3. Digital I/O

- Digital output: Draw a circuit to use an LED. Then write a code that will turn the LED on, wait for x milliseconds, then turn the LED off, wait for y milliseconds, and repeat indefinitely.
- Digital input: Draw a circuit to use a button to control an LED. Then write a code that will read and print the value of the button. Then add an if/else statement to print “LED on” or “LED off” on the serial monitor, depending on the value of the button.

4. User-defined functions

a. Write a user defined function that computes a simple computation, like a+bc (where a=2, b=4 and c=5). Return the result in the main code and print it once.
b. Same as above but a=2.5, b=4.2 and c=5.5.
c. Write a user defined function that reads the value of a sensor reading (plugged into pin A0) and returns it to the main code. The value is updated on the serial monitor indefinitely.
d. Same as above, but instead of printing the result indefinitely, print it 5 times.

5. Millis()

Given two integers a and b, write a code that will start a timer and print a value of (a+b) for the first second, then the value of (a-b) for the second second, then (a+b) for the third second, etc … to be repeated indefinitely.

Answers:

4. a.

```
void setup(){
  Serial.begin(9600);
  int a=2, b=4, c=5;
  int final_result;
  final_result = simple_computation (a,b,c);
  Serial.println(final_result);
}

void loop(){
}

int simple_computation (int d, int e, int f){
  int result;
  result = d+e*f;
  return result;
}
```

b.

```
void setup(){
  Serial.begin(9600);
  float a=2.5, b=4.2, c=5.5;
  float final_result;
  final_result = simple_computation (a,b,c);
  Serial.println(final_result);
}

void loop(){
}

float simple_computation (float d, float e, float f){
  float result;
  result = d+e*f;
  return result;
}
```

c.

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

void loop(){
  int sensor_pin = A0;
  int final_reading;
  final_reading = reading_function (sensor_pin);
  Serial.println(final_reading);
}

int reading_function (int pin){
  int result;
  result = analogRead(pin);
  return result;
}
```

d.

```
void setup(){
  Serial.begin(9600);
  int sensor_pin = A0;
  int final_reading;

  for (int i=1;i<=5;i++){
    final_reading = reading_function (sensor_pin);
    Serial.println(final_reading);
  }
}

void loop(){
}

int reading_function (int pin){
  int result;
  result = analogRead(pin);
  return result;
}
```

5.

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

void loop(){
  int a=5, b=12;
  static long start_time=0;
  long current_time = millis();

  if (current_time-start_time<=1000){
    Serial.print("a+b: "); Serial.println(a+b);
  }
  else if (current_time-start_time<=2000){
    Serial.print("a-b: "); Serial.println(a-b);
  }
  else{
    start_time = current_time;
  }
}
```