

---

# Engineering Technology (ENGR 101)

## Functions

---

# Arduino Functions

- What are functions good for?
  - structuring our thoughts (structured programming)
  - allowing us to re-use code, reducing work and reducing errors
- A C program can be modularised by functions
  - A big program can be broken down into a number of smaller ones

```

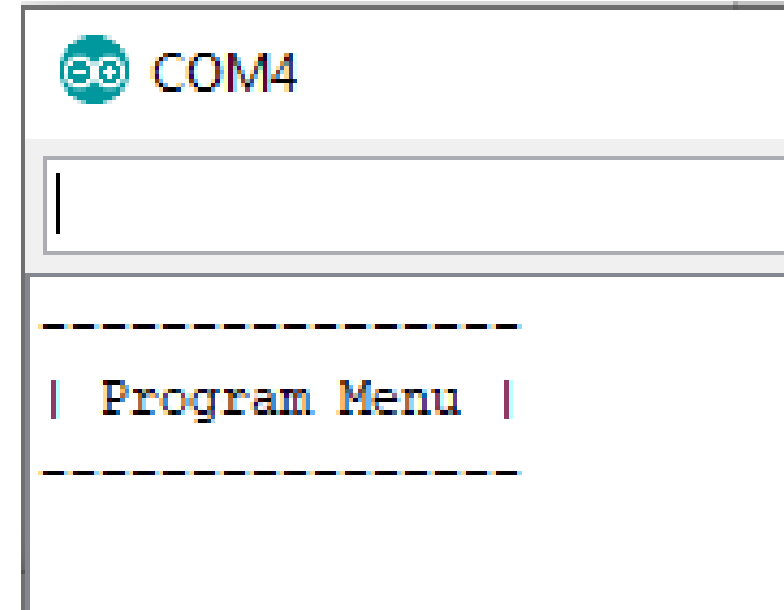
void setup() {
  Serial.begin(9600);
  DashedLine();
  Serial.println(" | Program Menu | ");
  DashedLine();
}

void loop() {
}

void DashedLine() {
  Serial.println("-----");
}

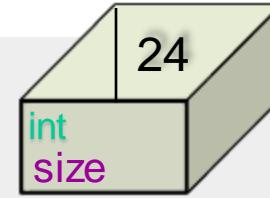
```

**Function is called here** (points to `DashedLine();` in `setup()`)  
**Function is called here** (points to `DashedLine();` in `setup()`)  
**Return type** (points to `void` in `loop()`)  
**Function Name** (points to `DashedLine` in `DashedLine()`)  
**Statement(s) that run when function is called** (points to `Serial.println("-----");` in `DashedLine()`)



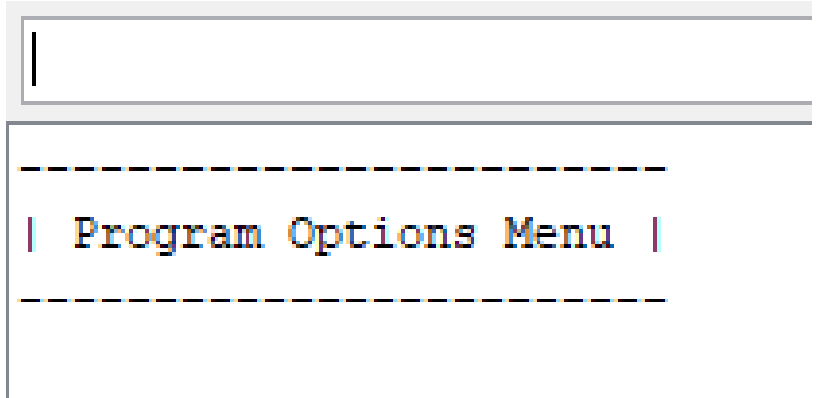
# Passing a Value to a Function

```
void setup() {
  Serial.begin(9600);
  int size = 24;
  DashedLine(size);
  Serial.println(" | Program Options Menu |");
  DashedLine(size);
}
```



Passing a value

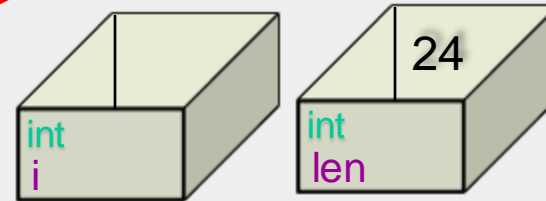
COM4



Arguments

```
void loop() {
}
```

```
void DashedLine(int len) {
  // draw the line
  for (int i = 0; i < len; i++) {
    Serial.print("-");
  }
  Serial.println("");
}
```

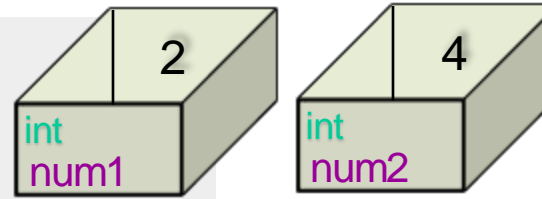


Parameters

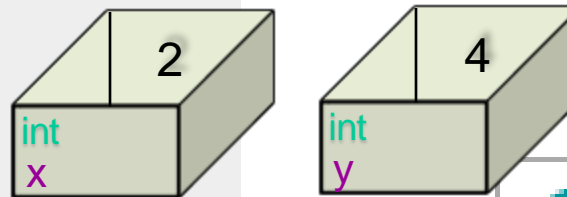
Special variables which are given values each time the function is called. Body of function can use the values in the parameters

# Another Example of a Function

```
void setup() {  
  int num1=2, num2=4;  
  Serial.begin(9600);  
  larger(num1, num2);  
}  
  
void loop() {  
}  
  
void larger(int x, int y) {  
  if (x>y) {  
    Serial.print(x);  
    Serial.println(" is larger");  
  }  
  else{  
    Serial.print(y);  
    Serial.println(" is larger");  
  }  
}
```



Passing values



COM4

4 is larger

# Function Calls with parameters

Function Definition: Like a pad of worksheets

```
void larger( int x, int y){  
       
    if (x>y) {  
        Serial.print(x);  
        Serial.println(" is larger");  
    }  
    else{  
        Serial.print(y);  
        Serial.println(" is larger");  
    }  
}
```

Calling a Function:

larger(2, 4);

- ⇒ get a “copy” of the function worksheet
- ⇒ copy the arguments to the parameter places
- ⇒ perform each action in the body
- ⇒ throw the worksheet away (losing all the information on it)

# Principle of good design

---

- Parameterizing a function makes it more flexible and general
  - Allows us to call the same function with different arguments to do the same thing in different ways
  - Allows us to reuse the same bit of code

# Returning a Value from a Function

```

void setup() {
  int num1 = 2, num2 = 4, lNum = 0;
  Serial.begin(9600);
  lNum = larger(num1, num2);
  Serial.print(lNum);
  Serial.println(" is larger");
}

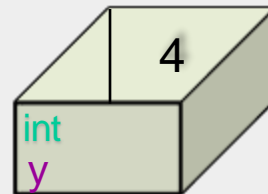
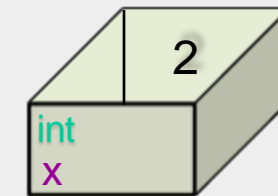
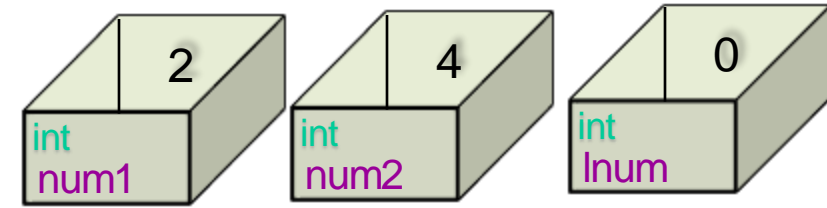
void loop() {
}

int larger(int x, int y) {
  if (x>y) {
    return x;
  }
  else{
    return y;
  }
}

```

**Return type** (points to `int` in the function signature)

**Passing values** (points to `num1` and `num2` in the `larger` call)



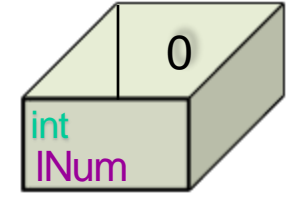
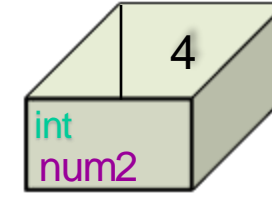
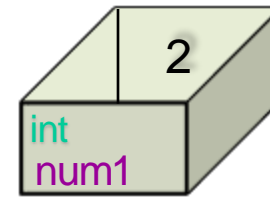
COM4

4 is larger

# Returning a Value from a Function

- What happens if we call the function:  
larger();

```
void setup() {  
  ✓ int num1 = 2, num2 = 4, lNum = 0;  
  
  ✓ Serial.begin(9600);  
  
  lNum = larger(num1, num2);  
  
  Serial.print(lNum);  
  Serial.println(" is larger");  
}
```





# Returning values

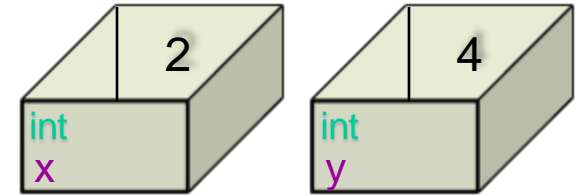
return value:

```
int larger(int x, int y){
```

```
    if (x>y) {  
        return x;  
    }
```

```
    else{  
        return y;  
    }
```

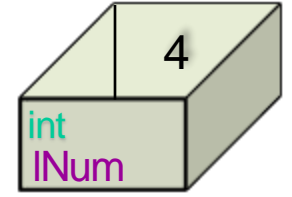
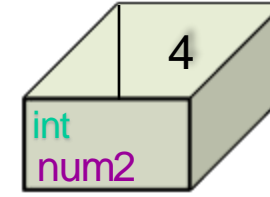
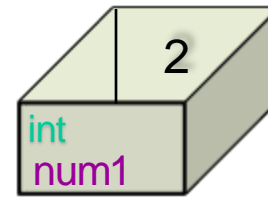
```
}
```



# Returning a Value from a Function

- What happens if we call the function:  
larger();

```
void setup() {  
  ✓ int num1 = 2, num2 = 4, lNum = 0;  
  
  ✓ Serial.begin(9600);  
  
  ✓ lNum = larger(num1, num2);  
  
  ✓ Serial.print(lNum);  
  ✓ Serial.println(" is larger");  
}
```



COM4

4 is larger

# Arduino Functions Definitions

```
returnType functionName(type parameter1, type parameter2, ...){  
    Function body  
    Optionally return value  
}
```

```
void setup() {  
    Serial.begin(9600);  
    DashedLine();  
    Serial.println(" | Program Menu |");  
    DashedLine();  
}  
void loop() {  
}  
void DashedLine() {  
    Serial.println("-----");  
}
```

```
int larger(int x, int y) {  
    if (x>y) {  
        return x;  
    }  
    else{  
        return y;  
    }  
}
```