Conditional statements are the heart of decision-making in programming.

**Suggested Prerequisites**
- Programming Intro

**Setup and Preparation**

**List of Materials**
- TI MSP430G2 LaunchPad
- Breadboard
- M/F Jumper Wires (4)
- M/M Jumper Wires (4)
- Pushbuttons (2)
- Resistors of small value (~1kΩ) (4)

**Prepare the circuit**
Design the following circuit (Figure 1).

*Figure 1: Conditional Lab Circuit*
Program 1: if statement

Create and upload the following program (Figure 2) to the LaunchPad.

```java
void setup() {
  pinMode(RED_LED, OUTPUT);
  pinMode(P2_3, INPUT);
}

void loop() {
  boolean pinState = digitalRead(P2_3);
  boolean buttonIsPressed = !pinState;
  boolean LEDstate = LOW;
  if (buttonIsPressed) {
    LEDstate = HIGH;
  }
  digitalWrite(RED_LED, LEDstate);
  delay(1);
}
```

Figure 2: if statement

Things to consider

- You can use a voltmeter to examine the voltage across the button when it gets pressed.
- The *Serial Monitor* can be used to examine the values of `pinState`, `buttonIsPressed`, and `LEDstate` (You’ll need to revise the code to enable this).
- There is a relationship between the values of `pinState` and the voltage across the button.
- Consider why the *NOT operator* `!` was included in this program.
- Consider how fast this program is repeating the code in the `loop()` function.
Program 2: Nested If statements

Create and upload the following program (Figure 3) to the LaunchPad.

```java
void setup() {
  pinMode(RED_LED, OUTPUT);
  pinMode(P2_3, INPUT);
  pinMode(P2_4, INPUT);
}

void loop() {
  boolean button1IsPressed = !digitalRead(P2_3);
  boolean button2IsPressed = !digitalRead(P2_4);
  boolean LEDstate = LOW;
  if (button1IsPressed) {
    if (button2IsPressed) {
      LEDstate = HIGH;
    }
  }
  digitalWrite(RED_LED, LEDstate);
  delay(1);
}
```

Figure 3: Nested if statements

Things to consider

- Consider the general effect of nesting if statements.
- You can analyze the program by examining the 3 variables button1IsPressed, button2IsPressed, and LEDstate, using the Serial Monitor.
- This program can be implemented with a single if statement, using Boolean operators.
- Information about Boolean operators can be found at the following link:
Program 3: If statements in sequence

Create and upload the following program (Figure 4) to the LaunchPad.

```c
void setup() {
  pinMode(RED_LED, OUTPUT);
  pinMode(P2_3, INPUT);
  pinMode(P2_4, INPUT);
}

void loop() {
  boolean button1IsPressed = !digitalRead(P2_3);
  boolean button2IsPressed = !digitalRead(P2_4);
  boolean LEDstate = LOW;
  if (button1IsPressed) {
    LEDstate = HIGH;
  }
  if (button2IsPressed) {
    LEDstate = HIGH;
  }
  digitalWrite(RED_LED, LEDstate);
  delay(1);
}
```

Figure 4: If statements in sequence

Things to consider

- Consider the general effect of putting if statements in sequence that do the same thing but are activated by different conditional statements.

- You can analyze the program by examining the 3 variables button1IsPressed, button2IsPressed, REDstate, and GREENstate using the Serial Monitor.

- This program can be implemented with a single if statement using Boolean operators.

- Information about Boolean operators can be found at the following link:
Program 4: else if statements

Create and upload the following program (Figure 5) to the LaunchPad.

```java
void setup() {
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(P2_3, INPUT);
  pinMode(P2_4, INPUT);
}

void loop() {
  boolean button1IsPressed = !digitalRead(P2_3);
  boolean button2IsPressed = !digitalRead(P2_4);
  boolean REDstate = LOW;
  boolean GREENstate = LOW;
  if (button1IsPressed) {
    REDstate = HIGH;
  }
  else if (button2IsPressed) {
    GREENstate = HIGH;
  }
  digitalWrite(RED_LED, REDstate);
  digitalWrite(GREEN_LED, GREENstate);
  delay(1);
}
```

Figure 5: else if statement

Things to consider

- Unlike `if statements` in sequence, `else if statements` are linked to their preceding `if statement`, and execution is mutually exclusive. You can determine which conditional statement has precedence in the event that they are both true.

- You can analyze the program by examining the 3 variables `button1IsPressed`, `button2IsPressed`, `REDstate`, and `GREENstate` using the `Serial Monitor`.

- You can try to explain the logical flow of the `if...else if` code structure with a block diagram.
Program 5: else statements

Create and upload the following program (Figure 6) to the LaunchPad.

```cpp
void setup() {
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(P2_3, INPUT);
}

void loop() {
  boolean buttonIsPressed = !digitalRead(P2_3);
  boolean REDstate = LOW;
  boolean GREENstate = LOW;
  if (buttonIsPressed) {
    REDstate = HIGH;
  }
  else {
    GREENstate = HIGH;
  }
  digitalWrite(RED_LED, REDstate);
  digitalWrite(GREEN_LED, GREENstate);
  delay(1);
}
```

Figure 6: else statement

Things to consider

- Similar to `else if statements`, `else statements` are linked to the preceding `if statement`.
- Notice that the else statement does not contain any conditional statement.
- You can analyze the program by examining the 3 variables `buttonIsPressed`, `REDstate`, and `GREENstate` using the `Serial Monitor`.
- You can try to explain the logical flow of the `if...else` code structure with a block diagram.
- It is also common to put an `else statement` after an `if...else if` code structure.
- Information about `if statements` and their variations can be found at the following link:
Review

After this lab, you should have a good understanding of the following topics. If you’re not sure about some of them, go back through the lab and try to find a good place to explore the topic.

You should be able to do the following

- Create programs that use *if statements* and its variants to execute code conditionally.
- Explain the logical flow of an *if statements* and its variants with block diagrams.
- Use *NOT, AND, and OR* Boolean operators to make more sophisticated conditional statements.
- Use the voltmeter and *Serial Monitor* to analyze and troubleshoot a circuit or program.

Application

Try applying the skills you learned from this lab. You can come up with your own project idea or try one from the flowchart.