# Make Several Decisions

Submitted by Andy Lindsay on Thu, 03/21/2013 - 16:34

Lesson edited to work with **CodeBlocks** IDE by Jeff La Favre 3/12/26

*(Updated 2013-08-08 for SimpleIDE 0.9.4 and its Learn folder's Simple Libraries and Examples) [SimpleIDE is the IDE for use with the robot. This lesson is edited so that we can use **CodeBlocks** as the IDE, which does not require a robot – J. La Favre]*

Make a Decision introduced the **`if`** statement, where a code block gets executed if it's **`if`** statement is true. If a program has multiple **`if`** statements that are true, more than one code block might get executed. But sometimes, you might only want *one* code block to execute based on a list of conditions.

For example, let's say we want one message when **`a`** is greater than **`b`**, a different message when **`a`** is less than **`b`**, and a third message if **`a`** is equal to **`b`**.

Another example you might see later is a robot with two contact sensors. It needs to back up and turn in different directions depending on whether both sensors are pressed, or just the left, or just the right.

- Start **CodeBlocks** (or the IDE you are using).
- Open the **File** menu and select **New**. Then select **Empty File**.
- Click the mouse in the text window of **CodeBlocks** and use the keyboard to enter the following text: **#include <stdio.h>**
- Open the **File** menu and select **Save file**, which opens a **Save file** dialog box.
- In the dialog box, open the drop-down labeled **Save as type** and select **C/C++ files**. In the **file name** slot enter this name for the file: **decision chain.** At the top of the dialog box there is a **Save in** slot, which determines where the file will be saved. Make sure you know the location where you are saving your file so that you can find it later. Now click the **Save** button to save your program file.

- Use your keyboard to type the text in the box on the next page into the text window of **CodeBlocks** under the first line of text you have already entered.
- Click the **Save file** button to save the code you just entered with keyboard.
- Examine the program and predict what result you think it will display with `a = 25` and `b = 17`. Repeat for `a = 17` and `b = 25` as well as for `a = 25` and `b = 25`.
- Run the program by opening the **Build** menu and selecting **Build and Run**. If there are no errors in the program, a new program window will open. Compare the actual output to your predicted output. Make sure to try all three combinations of `a` and `b` values.

```
int main()
{ //start of code block for main()
   int a = 25;
  int b = 17;
  printf("a = %d, b = %d\n", a, b);
  if(a > b)
  { //start of code block for if(a>b)
    printf("a is larger \n");   //if variable a is larger than variable b, then this line will execute
  } //end of code block for if(a>b)
  else if (a < b)
  { //start of code block for else if(a<b)
    printf("b is larger \n");      //if variable b is larger than variable a, then this line will execute
  } //end of code block for else if(a<b)
  else
  { //start of code block for else
    printf("a equals b \n");  //if variable b is not larger than a and a is not larger than b, then a must equal b, so execute this line
  } //end  of code block for else
} //end of code block for main()
```

## How it Works

The **`if...else if...else`** statement first checks if **`a`** is greater than **`b`** with **`if(a > b)`**. If it is, then the **`printf("a is larger \n")`** gets executed. An important point here is that it skips checking any of the other conditions in the statement and moves on to whatever code might be below the **`else{...}`**block. Now, if **`a`** is *not* greater than **`b`**, it does *not* execute **`printf("a is larger \n")`** and instead moves on to check the next condition: if **`a`** is less than **`b`** with **`if(a < b){...}`**. If that's true, it'll print that message instead. If that's not true either, the code will move on to the **`else`** condition, which is an optional catch-all if nothing else is true.

---

## Did You Know?

- You can add more **`else if`** statements between the first **`if`** and the last **`else`** catch-all.
- The **`else`** catch-all block is optional.

---

```
if(a > b)
{
  printf("a is larger \n");
}
else if (b == 1000)    //code  block below will execute if variable b is storing a value of 1000
{
  printf("WARNING, b is 1000 \n");
}
else if (a < b)
{
  printf(" b is larger \n");
}
else
{
  printf("a equals b \n");
}
}
```

## Try This

There are two `else if` conditions in the `if...else if...else` block on page 5. The first one checks for a special condition, which is when `b` equals 1000. In this case, the program just displays a warning message. However in other applications such as a robot or factory equipment, `if` might have lots more code for remedying that special condition. Now, if `b` is greater than `a`, but it's not equal to 1000, the code just displays the standard message.

- Open the **File** menu and select **Save file As** and save your project as **decision chain add else if**.
- Add the new `else if` code block in the excerpt on page 5.
- Change `b` to 1000 and build and run.
- Change `b` to 1001 and build and run the program. Did the program for both cases match your expectations?

## Your Turn

- Modify your program to display whether a number is positive, negative, or zero.