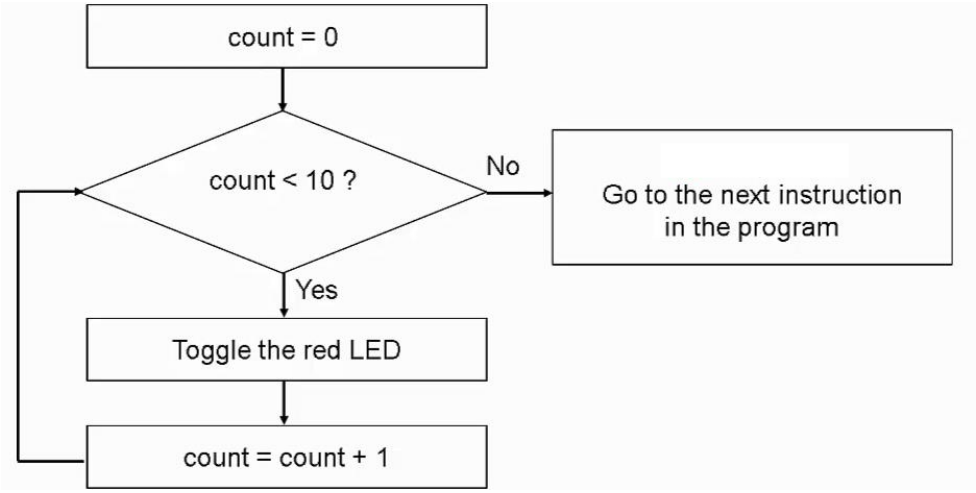
What Is a Flow Chart?

1. When creating a program you don't usually jump straight into writing it. Instead, you go through a design process.

   First, gather your ideas and decided what your constraints and requirements are. For example, let's think back to one of our previous programs that blinked the toggled the red LED ten times.

2. The next part of the design process when coding in C is to develop a model of your program. Many developers will use a flow chart for this step.

   A flow chart is a graphical representation of a program. For instance, consider the following flow chart.



3. This flow chart shows us that we start with a variable called **`count`** that is equal to **`0`**.

4. Next, the program "flows" to its next functional block. This block tests if the **`count`** variable is less than ten.

5. If **`count`** is less than 10, the program toggles the red LED, increments the **`count`** variable, and returns to test the variable again.

   If **`count`** is NOT less than 10, the program moves on to the next instruction.

6. In out next video and handout, we will formally introduce the **`while`** loop. Yes, we have been using already, but we'll take some time and go over it with some more details and examples.

   After we have been formally introduced to the **`while`** loop, you will see more examples of flow charts in the videos and handouts.

7. Now, we know that there are a lot of developers and programmers that don't like to use flow charts.

   It is true that they can take time to create and potentially can slow experts down.

   However, this is one of the reasons that I ask students to work with flow charts when they begin coding for the first time. It slows them down and makes them think about each individual step they are trying to perform.

   That being said – you don't have to use flow charts if you don't want to. In this class, no one is going to scold you for skipping this step. There are other organization tools out there for software developers that are commonly used such as pseudo code. But, we will use them in this course when we introduce new concepts because their graphical format do allow us to convey concepts relatively quickly.

All tutorials and software examples included herewith are intended solely for educational purposes. The material is provided in an "as is" condition. Any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for particular purposes are disclaimed.

The software examples are self-contained low-level programs that typically demonstrate a single peripheral function or device feature in a highly concise manner. Therefore, the code may rely on the device's power-on default register values and settings such as the clock configuration and care must be taken when combining code from several examples to avoid potential side effects. Additionally, the tutorials and software examples should not be considered for use in life support devices or systems or mission critical devices or systems.

In no event shall the owner or contributors to the tutorials and software be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.