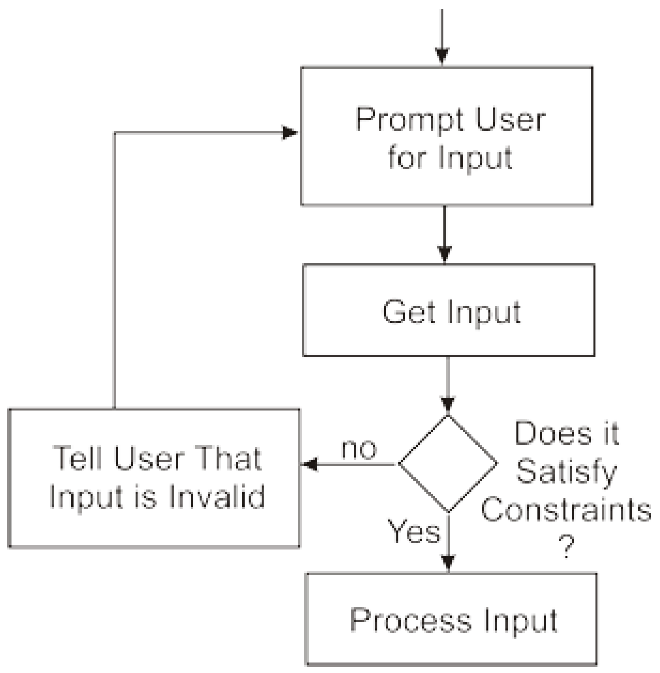
Testing Values to be Within a Range

Problem Outline

Most programs that get input from a user require that the data satisfy some constraints. Consider a simple game where the user has to guess a value between 1 and 10. Testing the user's input data to see if it lies within this range is a common way to use both branches and loops.

Solution

When gathering input from the user that must meet some constraints, a program needs to follow the steps illustrated in the activity diagram shown below:


Prompt User for Input
Get Input
Does it Satisfy Constraints ?
no
Yes
Tell User That Input is Invalid
Process Input

In this example, the program first prompts the user to input some data. If the input must satisfy some constraint, then you should tell this to the user. Then get the input from the user. Now test it to see if it satisfies the constraint. In our example problem, the input data must be between 1 and 10. A boolean expression of the form

```
    if (input >=1 && input <= 10)
```

would handle this test nicely. Note that since we want to print an error message if the constraint is **not** met, we could also have used the inverse condition

```
    if (input < 1 || input > 10)
```

If the test fails, the program should print a message to the user, and then go back and ask for the input again. The program should stay in this loop until the constraint is satisfied. Once the user input is validated, then the program can go on to process the data.

Coding the Solution

It should be obvious from the diagram that we need some kind of looping mechanism to code up the solution to this problem. What kind of loop would serve best? Notice that we always want to go through the body of the loop at least once. That is, we always want to get some input and test it. Recall that a do-while loop meets this criteria. It will perform the body of the loop first, and then test some condition to see if the loop should be executed again. Inside the loop, we want to

1. Prompt the user
2. Get the user's input
3. See if the input meets the given constraints
4. If it does not, print a warning message

The looping condition will test the input value against the constraint. The code to do this could thus be written as shown below:

```
    const int MAX = 10;
    ...
    do
    {
        Console.WriteLine("Enter a number between 1 and
{0}:", MAX);
        value = int.Parse(Console.ReadLine( ) );
        if( value < 1 || value > MAX)
        {
            Console.WriteLine("Your number isn't between
1 and {0}.", MAX);
        }
    }
    while (value < 1 || value > MAX);
```