At this point, you should feel pretty comfortable using Xcode. You should know how to open a project and how to edit a project’s source code. You should also feel comfortable running a project and (heaven forbid) fixing any syntax errors that may have occurred along the way.

On the programming side, you should recognize a function when you see one. When you think of a function, you should first think of main(), the function that gets called to start your program. You should remember that functions are made up of statements.

With these things in mind, you’re ready to explore the foundation of C programming: variables and operators. Variables and operators are the building blocks you’ll use to construct your program’s statements.

An Introduction to Variables

A large part of the programming process involves working with data. You might need to add together a column of numbers or sort a list of names alphabetically. The tricky part of this process is representing your data in a program, which is where variables come in.

Variables can be thought of as containers for your program’s data. Imagine a table with three containers sitting on it. Each container represents a different variable. One container is labeled cup1, one labeled cup2, and the third cup3.
Now imagine you have three plastic numbers. Place one number inside each of the three containers. Figure 4-1 shows a picture of what this might look like.

![Figure 4-1. Three cups, each one labeled and each with its own value](image)

Each cup represents a different variable.

Now imagine asking a friend to reach into each cup, pull out the number in each one, and add the three values together. You can ask your friend to place the sum of the three values in a fourth container created just for this purpose. The fourth container is labeled sum and can be seen in Figure 4-2.

![Figure 4-2. Four cups, one of which is the sum of the other three](image)

This is exactly how variables work. Variables are containers for your program’s data. You create a variable and place a value in it. You then ask the computer to do something with the value in your variable. You can ask the computer to add three variables together, placing the result in a fourth variable. You can even ask the computer to take the value in a variable, multiply it by two, and place the result back into the original variable.

Getting back to the example, now imagine that you changed the values in cup1, cup2, and cup3. Once again, you could call on your friend to add the three values, updating the value in the container sum. You’ve reused the same variables, using the same formula, to achieve a different result. Here’s the C version of this formula:

```
sum = cup1 + cup2 + cup3;
```