



Computer programming is fun, but it can also be frustrating. With time, patience, and practice, your knowledge will increase and your skills will improve. Your learning curve will increase more rapidly, however, if you develop good programming habits right from the start. Countless professional programmers before you have discovered efficient and effective ways to write and test code. In this lesson you will learn some of the habits of these highly successful programmers, and begin applying them yourself.

### **Habit No. 1 of Successful Computer Programmers: Analyze the Problem**

This seems like common sense, but many newbie programmers will read an assignment, launch their IDE, and immediately begin writing code. This is a huge mistake! Take time to understand and analyze the requirements of the task. Professionals will tell you that tremendous amounts of time are wasted when programmers don't have a good grasp of the task they are assigned to complete. So, when you are given an assignment, read it carefully. Make notes; ask yourself questions as we have done in previous lessons. Know what is expected so you stay on task and don't get distracted doing things that are not required. If you are unclear about the instructions for an assignment, ask for clarification. Programming is often hard, tedious work, but it is also a lot of fun. You will enjoy the experience more if you know what you are doing. You are probably very busy and don't have a lot of time to waste. So, if you don't have time to re-do an assignment, make sure you know what to do before you start!

## **Part 2**

### **Habit No. 2 of Successful Computer Programmers: Make a Plan**

A second common mistake beginning programmers make is working without a plan. This is *almost* as disastrous as a trapeze artist working without a net! At some point you will probably fall into this trap because a programming task may seem so simple that you can do it without a plan. But, as programs become bigger and more complicated, you will regret the "code first, then plan when it does not work" approach, because it is a huge waste of time. Don't give into temptation! A program is really just a list of instructions to accomplish a task written in a language the computer understands. For now, you can simply think of your plan to write a program as a To Do List, the orderly steps necessary to perform the task. Programmers refer to this as an algorithm, a list of instructions to solve a problem. If you were asked to build a house, would you immediately start hammering and sawing? Hopefully you would first develop a plan (e.g., a blueprint). Taking the time to write down the steps to solve a problem before you start any program is a key habit of highly successful programmers. Planning will save you tremendous amounts of time. In future lessons we will develop some more advanced ways to plan your programs, but the analogy of a To Do List will work well for now.

## **Part 3**

### **Habit No. 3 of Successful Computer Programmers: Test Often**

Accept the fact that you are going to make mistakes (lots of them) when writing programs. Everybody does. The sooner you can find your errors the better, so highly successful programmers test their code often. The trick is to find little mistakes that are easy to fix quickly, before they grow into big mistakes that are more time-consuming to find and repair. If you were building a house, would you wait until the roof was installed before you asked the inspector to check the plumbing? Big mistakes can be minimized by writing a few lines of code and then compiling. If the compiler finds an error, fix it immediately. Then run the program. If there are runtime errors, fix those before writing any new code. Once you have eliminated compiler and runtime errors, write a few more lines and test again. Any errors found since the last error-free compilation and run can probably be traced to the most recently added code. Testing your programs frequently is a big time saver.

## **Part 4**

### **Habit No. 4 of Successful Computer Programmers: Embrace Your Mistakes**

A major reason that some students give up on learning computer programming is the frustration of dealing with all the bugs. It is human nature to want to perform a task correctly, and to do it right on the first attempt. Failure is an uncomfortable feeling. But, you have three choices: give up, learn from your mistakes, or get over it.

- Giving up does not mean quitting. Take a break. Do something else to take your mind off the bug(s) you can't seem to fix. Too many students will sit for hours trying to fix one problem, but not really thinking about it, just making lots of almost random changes ... hoping something, anything will eventually work. Long before that point, it's time to step away from the computer! Let your subconscious deal with the bug for awhile. Tackle the problem again when you are fresh. You can also ask for assistance from your instructor after you have made a good faith effort.
- Learning from your mistakes is actually a good strategy. Although most people would rather hide a mistake than admit it, you should look at every mistake as a learning opportunity. Thomas Edison is said to have made 10,000 failed attempts to create the light bulb, yet from every mistake, he learned how *not* to make a light bulb. You are going to make more mistakes in this course than probably any other course you ever take. Every bug you fix will teach you something to avoid, if you learn from experience, and you are about to begin gaining a lot of experience at fixing bugs in your code.
- So, get over it. Programmers enjoy the destination, not necessarily the journey. The feeling you get once you complete a particularly difficult assignment is hard

to describe. But basically it is a feeling of satisfaction. The compiler has thrown error after error at you, and you triumphed! On top of that, you may have created a pretty neat program! So enjoy the sensation, it will only last until you start another assignment, but the anticipation of another success will keep you going.

There are many more Habits of Highly Successful Computer Programmers to be covered during the course, but the bottom line is that improving any skill takes practice and developing good habits. The first time you shot a basketball, wrote an essay, danced the Salsa, tried to speak a foreign language, rode a horse, etc., chances are you did not do it very well. It takes observation and practice to build any skill. In fact, we often learn best by mimicking someone who is successful as our own skill and confidence, grows. Often we just go through the moves until everything suddenly falls together and we understand.

Begin developing good programming habits as you go through the course and you will greatly reduce the amount of time it takes to complete assignments.