

1. **Analyze the Problem** Figure out exactly what the problem to be solved is. Try to understand as much as possible about it. Until you really know what the problem is, you cannot begin to solve it.
2. **Determine Specifications** Describe exactly what your program will do. At this point, you should not worry about *how* your program will work, but rather about deciding exactly *what* it will accomplish. For simple programs this involves carefully describing what the inputs and outputs of the program will be and how they relate to each other.
3. **Create a Design** Formulate the overall structure of the program. This is where the *how* of the program gets worked out. The main task is to design the algorithm(s) that will meet the specifications.
4. **Implement the Design** Translate the design into a computer language and put it into the computer. In this [class] we will be implementing our algorithms as [Java] programs.
5. **Test/Debug the Program** Try out your program and see if it works as expected. If there are any errors (often called *bugs*), then you should go back and fix them. The process of locating and fixing errors is called *debugging* a program. During the debugging phase, your goal is to find errors, so you should try everything you can think of that might “break” the program...
6. **Maintain the Program** Continue developing the program in response to the needs of your users. Most programs are never really finished; they keep evolving over years of use.

¹Quoted from *Python Programming*, 2nd edition, by John Zelle, Franklin, Beedle, and Associates Inc., 2010, Page 28