

Six Steps Handout

Six Steps to Creating a Computer Program

When creating new programs to solve problems for us, it helps to follow a consistent six step process. This process should work for most problems that you want to solve, and including the lab assignments this semester.

1. **Analyze the Problem:** Make sure you understand what it is that you want to solve.
2. **Determine the Specifications:** What do you want the result of your program to be (should it create something)? What inputs do you need to create that result?
3. **Create a Design:** Sketch out the steps needed for your solution that you could explain to a friend who knows nothing about computer science.
4. **Implement the Design:** Write a program in your favorite programming language (obviously Python) that follows your designed solution.
5. **Test/Debug the Program:** Check whether your program produces the expected outputs for different inputs. If it doesn't work, use the wrong outputs to try to guess what might be wrong with your implementation.
6. **Maintain the Program:** Make changes as needed to solve new or different problems, reusing your past designs.

Practice Problem

To practice this process, consider the following problem: we want to write a program that can solve Sudoku puzzles for us.

1. **Analyze the Problem.** What is the problem asking you to solve? What are the rules of Sudoku puzzles?

		3		2		6		
9			3		5			1
		1	8		6	4		
		8	1		2	9		
7								8
		6	7		8	2		
		2	6		9	5		
8			2		3			9
		5		1		3		

2. Determine the Specifications. What should the inputs be to your program? What should it output?
3. Create a Design. How might you go about solving this problem? What are the different steps your program will need to perform?
4. We will implement a Sudoku solver as an in-class exercise later this semester using the tools and concepts we learn along the way. Stay tuned!
5. As a thought exercise, think about how you might test a Sudoku solving program. What kinds of inputs would you give it, and how would you check the resulting outputs?
6. It turns out that maintaining our Sudoku solving program could involve adapting it to solve other problems that are very closely related: (a) creating the final exam schedule at Oberlin College, (b) deciding how cell phones connect to nearby cell towers, and (c) coloring all of the different states or countries on a map. We discuss this in more detail in CSCI 364 (Artificial Intelligence).