Notes compiled by TA Jiamen Chen

goal of debugging: not to eliminate one bug but for a bug-free program

two most powerful debugging tools are print statement + reading

be systematic: reducing the search space to localize the source of the problem instead of a random fashion

study program text, and ask how could it produce this result

Scientific Method for Debugging

first start by studying the available data: test results, all the test results(including those that worked - understanding why it worked on some but not others might provide some insight)

study the program text with skeptical eye: keep in mind that you don't understand it - that's why you made the mistake

form hypothesis

design and run a repeatable experiment: the experiment must have the potential to refute the hypothesis

useful intermediate results

Have an expected result: think in advance what should be the results, what the program does.

find the simplest input that can cause the problem

General Principles:

1) the bugs are probably not where you think they are: or you would have found it

common bugs: reversed order of arguments, wrong spelling, initialization(inside/outside the loop makes a difference), aliasing

2) Keep a record of what you tried: avoid doing the same thing repeatedly, do not waste time trying the same thing again

3) Reconsider assumption

4) Debug code not comments

5) get help from others, explain your programs to others

6) walk away to get a fresh mind, might find something that you did not notice before

7) before make changes to a program, think through what the change might break, what the change does, would it remedy all the problems.

one thing to keep in mind: code should not always grow, sometimes you should really tidy things up, and may accidentally find the bug

8) make sure that you can revert: keep a copy of old versions so that if your changes make the code worse, you still have the access to the older versions.