Coding and Problem Solving

• Practice
Coding: Problem Solving

- Practice, practice, practice
Coding: Problem Solving

- Practice, practice, practice
Coding: Problem Solving

- Practice, practice, practice
- Patience
- ...

Coding: Problem Solving

- Practice, practice, practice
- Patience
- ...
- Profit!
Coding: Problem Solving

- Practice, practice, practice
- Patience
- ...
- Profit Joy/Knowledge!
How to design code
How to design code

- Analyze the problem
How to design code

- Analyze the problem
- Research solutions
How to design code

- Analyze the problem
- Research solutions
- Design an algorithm
How to design code

- Analyze the problem
- Research solutions
- Design an algorithm
- Writing the code
Sample Problem:

Write a program that will reverse the order of inputted strings.
Analyze the problem
Analyze the problem

• What am I trying to do?
Analyze the problem

- What am I trying to do?
- Do I understand the problem?
Analyze the problem

- **What am I trying to do?**
- Do I understand the problem?
- What key concepts are involved?
Analyze the problem

• **What am I trying to do?**
  • Do I understand the problem?
  • What key concepts are involved?
  • Can I break this problem down to sub-tasks?
Research Solutions
Research Solutions

• How can I do it?
Research Solutions

• How can I do it?
  • Try to visualize what a solution would be
Research Solutions

• How can I do it?
  • Try to visualize what a solution would be
  • What key subtasks will we need?
Research Solutions

• **How can I do it?**
  • Try to visualize what a solution would be
  • What key subtasks will we need?
  • What libraries might help with this?
Design an algorithm
Design an algorithm

- What is the best/fastest/easiest way?
Design an algorithm

- What is the best/fastest/easiest way?
- Should I use iteration/recursion/something else?
Design an algorithm

• What is the best/fastest/easiest way?
• Should I use iteration/recursion/something else?
• What data structures should I use?
Design an algorithm

- What is the best/fastest/easiest way?
- Should I use iteration/recursion/something else?
- What data structures should I use?
- Break the program into simpler sub problems!!!
Design an algorithm

- What is the best/fastest/easiest way?
- Should I use iteration/recursion/something else?
- What data structures should I use?
- **Break the program into simpler sub problems!!!**
- Write some Pseudo code
Design an algorithm

• What is the best/fastest/easiest way?
• Should I use iteration/recursion/something else?
• What data structures should I use?
• Break the program into simpler sub problems!!!
• Write some Pseudo code
• If we get stuck... Break the problem into simpler sub problems!!!
Write the code
Write the code

● If you start with pseudo code, it's easy to convert it into real code!
Write the code

- If you start with pseudo code, it's easy to convert it into real code!
- Write function documentation first
Write the code

- If you start with pseudo code, it's easy to convert it into real code!
- Write function documentation first
  - No, seriously!
Write the code

- If you start with pseudo code, it's easy to convert it into real code!
- Write function documentation first
  - No, seriously!
- Debug as often as needed
  - print statements
  - the python debugger
Re-design
Re-design

- Trial and error is often a big part when starting to work on a problem.
Re-design

• Trial and error is often a big part when starting to work on a problem.

  writeCode()
  while True:
    rewriteCode()
Re-design

- Trial and error is often a big part when starting to work on a problem.
  
  ```python
  def writeCode()
      while True:
          rewriteCode()
  ```

- Use debugging statements while working out bugs
Summary... Coding Process

- **Process:**
  - Practice...
  - Patience...
  - **Experience joy in solving problems!**

- **Steps:**
  - Analyze the problem
  - Research the problem
  - Design a solution algorithm
  - Write the code
  - Fix the code/testing