

Problem Decomposition

Breaking complicated problems into manageable pieces

Think before you code



Make sure you understand the problem

- ▶ Restate it in your own words
- ▶ Know what the desired inputs and outputs are
- ▶ Ask questions
 - ▶ Of whomever is telling you what to program
 - ▶ This will typically be you



Break the problem down

- ▶ Divide the problem into a few large steps
- ▶ Write these down
 - ▶ On paper
 - ▶ As comments in a file



Break the problem down further

- ▶ If one or more of the steps from your first break down is still large/confusing then break it down again
- ▶ Iterate



Code one small piece at a time

- ▶ Pick one piece and think about the way to implement it
- ▶ Write this code/query
- ▶ Test it... on it's own
 - ▶ You might need to tweak your code to do this
- ▶ Fix any problems
- ▶ Lather, rinse, repeat



Code things that are simpler than you need

- ▶ If you need a single function to do something complicated
- ▶ And you're not sure how to do it
- ▶ Write a simpler variant first
 - ▶ Experiment
 - ▶ Make sure that it's working
 - ▶ And that you understand it
 - ▶ And then modify it to make it more complicated



Problem Decomposition Steps

1. Understand the problem
2. Break the problem down into a few large pieces
3. Break the pieces into codeable chunks
4. Code one chunk at a time
 1. Think about it
 2. Write it
 3. Test it (on it's own)
 4. Fix any problems



Simplifying functions

- ▶ If functions are confusing start without a function
- ▶ Then convert it into a function



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used
3. Replace the specific values in your example with the variables
4. Use the names of those variables in the function definition
5. Indent
6. Remove the initial definition of the variables
7. Return you answer
8. Call the function



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used
3. Replace the specific values in your example with the variables
4. Use the names of those variables in the function definition
5. Indent
6. Remove the initial definition of the variables
7. Return you answer
8. Call the function

**Write a function to
add two numbers**



From commands to functions

1. Write a specific example of what the function should do

**Write a function to
add two numbers**

`answer = 5 + 2`



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used

**Write a function to
add two numbers**

```
num1 = 5
```

```
num2 = 2
```

```
answer = 5 + 2
```



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used
3. Replace the specific values in your example with the variables

**Write a function to
add two numbers**

```
num1 = 5
```

```
num2 = 2
```

```
answer = num1 + num2
```



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used
3. Replace the specific values in your example with the variables
4. Use the names of those variables in the function definition

Write a function to add two numbers

```
def add(num1, num2):  
    num1 = 5  
    num2 = 2  
    answer = num1 + num2
```



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used
3. Replace the specific values in your example with the variables
4. Use the names of those variables in the function definition
5. Indent

Write a function to add two numbers

```
def add(num1, num2):  
    num1 = 5  
    num2 = 2  
    answer = num1 + num2
```



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used
3. Replace the specific values in your example with the variables
4. Use the names of those variables in the function definition
5. Indent
6. Remove the initial definition of the variables

**Write a function to
add two numbers**

```
def add(num1, num2):  
    answer = num1 + num2
```



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used
3. Replace the specific values in your example with the variables
4. Use the names of those variables in the function definition
5. Indent
6. Remove the initial definition of the variables
7. Return you answer

**Write a function to
add two numbers**

```
def add(num1, num2):  
    answer = num1 + num2  
    return answer
```



From commands to functions

1. Write a specific example of what the function should do
2. Create variables to hold the specific values you used
3. Replace the specific values in your example with the variables
4. Use the names of those variables in the function definition
5. Indent
6. Remove the initial definition of the variables
7. Return you answer
8. Call the function

Write a function to add two numbers

```
def add(num1, num2):  
    answer = num1 + num2  
    return answer  
myanswer = add(5, 2)
```

