Translation from Problem to Code in Seven Steps

Susan H. Rodger
Duke University
Durham, NC USA

Andrew D. Hilton
Duke University
Durham, NC USA

Genevieve M. Lipp
Duke University
Durham, NC USA

May 17, 2019
CompEd 2019
Chengdu, China
From Duke University, Durham, NC USA

Duke University Chapel

CompEd 2019

My cats
Novice programmers struggle!
problem $\rightarrow$ code

- How to start?
- Write an algorithm
- How to start that?
Novices Struggling (1)...

• First year students don’t know how to program at the end of the course!
  • ITiCSE Working Group 2001

• Over 200 students, average score 21%
Novices Struggling (2)...

• First year students weak in predicting the outcome of short code segments!
  • ITiCSE Working Group 2004
  • A multi-national study of reading and tracing skills in novice programmers, SIGCSE Bulletin 36, 4 (Dec 2004), p. 119-150

• Fragile grasp of skills needed for problem solving
Practices for Learning Programming

- Support to learners
  - Peer instruction, Pair Programming
- Focus on subject matter
  - Media Computation
- Drag-N-Drop environment
  - Alice, Scratch

- None in understanding HOW to develop an algorithm
Problem Solving Defined Many Ways

• Computational Thinking
  • Abstraction, algorithmic thinking, decomposition, pattern recognition

• Problem-based learning (PBL)
  • Seven steps: 1) examination of the case, 2) identification of problem, 3) brainstorming, etc.

• Others
  • All are very abstract!
Stuck on solving a problem? Don’t know where to start?

• Use the 7 step process!
Problem Solving to Code – Steps 1-4

1. Work small examples by hand
2. Write down what you did in words (algorithm)
3. Find Patterns (generalize algorithm)
4. Work another example by hand (algorithm work? If not, go back to 3, or 1)
Problem Solving to Code – Steps 5-7

5. Translate to code
6. Test several cases
7. Debug **failed** test cases
Problem Solving to Code

7 Step Process

1. Work small examples by hand
2. Write down what you did in words (algorithm)
3. Find Patterns (generalize algorithm)
4. Work another example by hand (algorithm work? If not, go back to 3, or 1)
5. Translate to code
6. Test several cases
7. Debug failed test cases
Example using Seven Steps

Problem Statement

Sabrina needs to be able to reorganize names into the last name first and she wants to abbreviate any middle names with the first letter and a period. She respects middle names that are a single letter and does not abbreviate them.

Write function modify that given a name returns the name with the last name first, followed by a comma, followed by the first name (if any), followed by the first letter of each remaining/middle name with a period after each letter. If a middle name is a single letter, do not abbreviate it/follow it by a period.
Examples

1. 
   name = "Prince"
   returns "Prince"
   There is only one name.

2. 
   name = "Thomas Narten"
   returns "Narten, Thomas"
   There is no middle name.

3. 
   name = "Liz A Bo To Joe"
   returns "Joe, Liz A B. T."
   All the middle names are abbreviated except A which was of length 1
Step 1) Work small example by hand

• **name** is “Moe Jess Bo Lu Yue”
• **first** is “Moe”
• **last** is “Yue”
• **middle** is “Jess Bo Lu”
  • “Jess” gives us “J.”
  • “Bo” gives us “B.”
  • Join together “J. B.”
  • “Lu” gives us “L.”
  • Join together “J. B. L.”
• Last, First Middle: “Yue, Moe J. B. L.”
Step 2) Describe in words what you did

• Name is: “Moe Jess Bo Lu Yue”
• Determine first name: “Moe”
• Determine last name: “Yue”
• Determine all middle names: “Jess Bo Lu”
  • Look at first word in middle: “Jess”
  • newMiddle is “J.”
  • Look at second word: “Bo”
  • NewMiddle is “J. B.”
  • Look at third word: “Lu”
  • NewMiddle is “J. B. L.”
• Put together last, first and newMiddle
Step 3) Find Patterns (Generalize)?
Don’t see it? Work another example

• Name is: “Sue Mo Lucy Lo So Fa Ti”
• Firstname is “Sue”, Lastname is “Ti”
• Middle is “Mo Lucy Lo So Fa”
  • “Mo”, newMid is “M.”,
  • “Lucy”, newMid is “M. L.”
  • “Lo”, newMid is “M. L. L.”
  • “So”, newMid is “M. L. L. S.”
  • “Fa”, newMid is “M. L. L. S. F.”
• Put together: “Ti, Sue M. L. L. S. F.”
Step 3) Find Patterns (Generalize)

- Name is: “Moe Jess Bo Lu Yue”
- Firstname is first word
- Lastname is last word
- Middle is string of all the middle words
- Initialize newMiddle
- For each word in middle:
  - Add first letter of word, period and blank to newMiddle
- Build new string:
  - lastname, firstname newMiddle
- Return answer
Step 4) Work another example by hand using your algorithm

• Name = “Jo Flo Bo Yup”
• Firstname = “Jo”
• Lastname = “Yup”
• Middle = “Flo Bo”
• newMid = “”

• For word in Middle:
  • newMid = “F.” (first time thru loop)
  • newMid = “F. B.” (second time thru loop)
• Answer = “Yup, Jo F. B.”
Step 5) Translate to Code

• Firstname is first word
  
  pos = name.find(" ")
  first = name[:pos]

• Lastname is last word

  rpos = name.rfind(" ")
  last = name[rpos+1:]

• Middle is string of all the middle words
Step 5) Translate to Code

• **Firstname is first word**
  
  \[
  \text{pos} = \text{name.find}(\text{" "} )
  \]
  
  \[
  \text{first} = \text{name}[\text{pos} : ]
  \]

• **Lastname is last word**
  
  \[
  \text{rpos} = \text{name.rfind}(\text{" "} )
  \]
  
  \[
  \text{last} = \text{name}[\text{rpos} + 1 : ]
  \]

• **Middle is string of all the middle words**
  
  \[
  \text{middle} = \text{name}[\text{pos} + 1 : \text{rpos}]
  \]
Step 5) Translate to Code

• Initialize newMiddle

• For each word in middle:
  • Add first letter, period and blank to newMiddle

• Build new string:
  • lastname, firstname newMiddle

• return the answer
Step 5) Translate to Code

• Initialize `newMiddle`
  `newMiddle = ""`

• For each word in `middle`:
  • Add first letter, period and blank to `newMiddle`
    for word in `middle.split()`:
      `newMiddle = newMiddle + word[0] + ". "`

• Build new string:
  • `lastname, firstname` `newMiddle`
    `answer = last + ", " + first + " " + newMiddle`

• return the answer
  `return answer`
Step 6) Test Several Cases

• Does our algorithm work for?
• Name = “Felicia Mary Moffet”
• Name = “Simon”
• Name = “Ronald McDonald”

• Need to handle special cases
Step 6) Test Several Cases

- Does our algorithm work for?
  - Name = “Felicia Mary Moffet”       YES
  - Name = “Simon”                    NO
  - Name = “Ronald McDonald”          NO

- Need to handle special cases
  - One or two words
Step 7) Debug Failed Test Cases

• How do you debug? Some tips
  • Isolate where the problem is
  • You think your code is correct, but is it?
  • Print out the value of variables.
  • Break code apart and print
  • Print, print, print
  • Identify your output
  • OR put function in Python tutor and call it on an example
Lab on 7 steps - Lego pattern
Lab on 7 steps - Lego pattern

- Given a series of boards with lego pattern
  - Grows with one lego each time
- What is the pattern with one more lego?
- Write a general algorithm for N legos
- Give your algorithm to another group to try
- Must be precise!

- Practicing Steps 1-4
Courses using 7-steps

• CompSci 101, Duke University
  • Intro programming majors, Python
  • 8 of 26 lectures (30%)
  • Walked through a problem with 7 steps in lecture
  • Lego lab

• ECE 551, Duke University
  • Intro programming grad students, C, C++
  • Flipped classroom, watch videos
  • 19 of the videos use Seven Steps
  • Lego Lab
Courses using 7-steps

• Online Coursera Courses
  • Java Programming and Software Engineering Fundamentals Specialization
  • Introduction to Programming in C Specialization
  • Both courses the 7 steps are used heavily in the videos
Student Anecdotes

• From CompSci 101
  • “I just want to tell you that I tried the seven step method, and I worked on all of my code for one or two hours before I even looked at the computer. AND IT WORKED! I got all my code right on the first try! For the first time ever, I don’t have to go to the help lab ...”
Student Anecdotes

• From Coursera course
  • “I have been programming for a couple of years. Learned from so many resources but none said how to write the algorithm, they just say you should write your algorithm first. The steps illustrated here are beautiful and definitely help to understand how to decompose a problem.”
Survey both Duke courses

- Course A is ECE 551
- Course B is CompSci 101
Reported Importance
Conclusions

• Seven Steps provides step-by-step approach
  • Includes how to devise the algorithm
• Needs to be integrated in throughout the course
• Works on small problems
  • Larger problems use it on components
• Future work – study with 7 steps
Questions?

Diagram:

(1) Work an instance yourself

(2) Write down exactly what you just did

(3) Generalize your steps from (2)

(4) Test your steps

Can't find pattern

Step 5: Translate to code

Step 6: Test program

Implementation problem

Program is incorrect

Program appears correct

Success!

Step 7: Debug program

Compea 2019 35