Compsci 101
7-Steps
Part 1 of 3

Susan Rodger
January 28, 2021
Join Duke Mailing lists
compsci@duke.edu

• Mailing list about
  • Jobs, internships, research positions
  • Events related to computer science

• How to join:
  • Go to: lists.duke.edu
  • Be sure to authenticate
  • Add compsci@duke.edu

• BE IN THE KNOW ABOUT COMPSCI!
C is for …

• Computer Science and Computing
  • It’s what we do
• Collaboration
  • Review the policy
• Cookies
  • Good for the web and for …
• CSV
  • Comma Separated Values: Data
PFTD

• 7 steps of programming
• Functions
• APTs
In Python version 3 print is a function

- Functions have parentheses
  - Arguments are provided in parentheses
  - We can `print(3+5)` or `print("hello")` or ...
  - What is returned by print?

- When there is no return value...
  - None is returned, it has no representation
  - Its type is `NoneType`
- Note: in python version 2, print is NOT a function, looks different
APTs in 101 and 201

• Algorithm Problem-solving and Testing
  • Algorithm that’s Automatically Tested
  • In use at Duke since 2003, million+ APTs solved

• Given a problem statement
  • Read, think, plan on paper …
  • Write a function to solve the problem
  • Submit the code for testing, debug if necessary
The Seven Steps
Programming Process: High-level

• First part: devise the algorithm
  • The meta-problem solving piece
  • Big/complex enough to be 4 steps (more shortly)
The Seven Steps
Programming Process: High-level

- After devising the algorithm, translate to code
  - Plan first, then code
  - Bridge analogy: blue prints, then construction
  - Essay analogy: outline, then prose
The Seven Steps
Programming Process: High-level

- Next test our program
  - Testing important, often under-taught skill
The Seven Steps
Programming Process: High-level

- Ideally would be correct first time; may need to debug
  - Identify problem (with science!)
  - Return to appropriate prior step to fix the problem
The Seven Steps
Programming Process: High-level

- Work through cycle until program works
Steps 1—4: Devise Algorithm

- Steps 1—4: devise the algorithm
  - Learn to do this well, be an excellent programmer
  - Language: does not matter
Compsci 101
7-Steps
Part 2 of 3

Susan Rodger
January 28, 2021
Steps 1—4: Example:
Calculate the average of two numbers

• Step 1: $2 + 5 = 7$, $7/2 = 3.5$
• Step 2:
  • Add $2 + 5$ and get 7
  • Divide 7 by 2 and the result is 3.5
Steps 1—4: Example: Calculate the average of two numbers

• Step 3:
  • Two variables num1 and num2
  • Add the two numbers together:
    result is num1 + num2
  • Divide the result by 2 and you have the answer
    answer is result / 2
Steps 1—4: Example: Calculate the average of two numbers

- Step 4: Try a different example
  - Use 8 and 6, num1 is 8, num2 is 6
  - Add the two numbers together:
    result is num1 + num2, is 14
  - Divide the result by 2 and you have the answer
    – Answer is result/2, which is 7

- IT WORKS!
Step 5: let’s convert it to code!

• Go to Pycharm
• We will also:
  • Test it – Step 6
  • Debug it – Step 7
Compsci 101
7-Steps
Part 3 of 3

Susan Rodger
January 28, 2021
APT: Write a Python Function

• We def(ine) functions in Python
  • Use indentation to create body of the function
  • Calling function is different than writing function

```python
def inch2centi(inches):
    return 2.54*inches

length = inch2centi(72)

def pluralize(word):
    return word + "es"

word = pluralize("fish")
```
Understanding Execution

• Using PythonTutor: http://pythontutor.com
  • How are functions defined?
  • Where does execution begin?
  • What is the global frame?
  • What is a local/function frame?