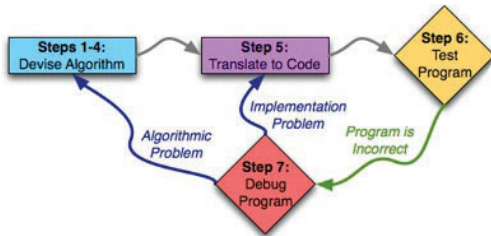


Compsci 101

7-steps, Functions, Order of Execution



Susan Rodger
September 6, 2022

```
Specification

filename: Laundry.py

def minutesNeeded(m):
    """
    Return integer number of minutes
    """
```

C is for ...



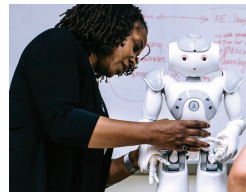
- **Computer Science and Computing**
 - It's what we do
- **Collaboration**
 - Review the policy
- **Cookies**
 - Good for the web and for ...
- **CSV**
 - Comma Separated Values: Data

Ayanna Howard

- Educator, Researcher and Innovator
- Professor & Chair of the School of Interactive Computing, Georgia Tech
- Now Dean of Engineering at The Ohio State University!
- Robotics – Robots and Bias, Robots changing lives of children with disabilities, Robots beyond part of the family
- Top 50 U.S. Women in Tech, Forbes, 2018



I believe that every engineer has a responsibility to make the world a better place. We are gifted with an amazing power to take people's wishes and make them a reality.



Announcements

- Lab 01 Friday,
 - Complete Prelab before going to lab
- APT-1 out today, due Thursday, September 15
- Assignment 0 due Thursday, September 8
- Sakai quizzes on readings due 10:15am on date due
 - Get three tries, score highest score
 - First two weeks we allow you to submit late
 - First 5 quizzes turn off, 10:15am Sept 13
- Read Ed Discussion Every Day – You will learn things!
- Reminder: Ed Discussion back channel in lecture!

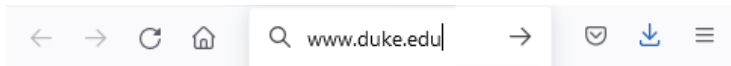
PFTD

- Functions
- Order of execution
- 7 steps of programming
- APTs
- Testing and Submitting APTs

What is a Function?

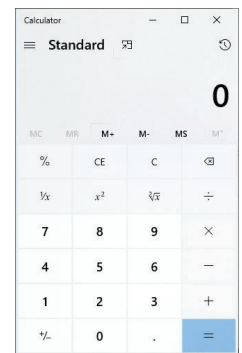
- **Function has:**
 - Name
 - Maybe inputs
 - Processes or calculates something
 - Has a result

Functions in the Real World: URL in webpage



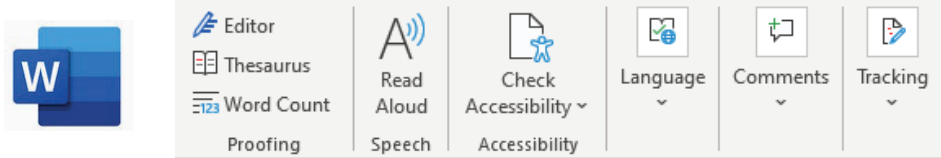
- **Function has:**
 - Name: "Search"
 - Input: www.duke.edu
 - Calculates:
 - Returns back:

Functions in the Real World: calculator



- **Function has:**
 - Name: calculator
 - Input: number(s), operator
 - Example: 25, squareroot
 - Calculates:
 - Returns back:

Functions in the Real World: Counting words in Microsoft Word



- **Function has:**
 - Name:
 - Input:
 - Calculates:
 - Returns back:

Built-in Python Function – len() already exists, you use it

- **len()** function
- **Function has:**
 - Name: `len`
 - Input: `a string`
 - Calculates: `number of characters in string`
 - Returns back: `number`

Examples:

```
x = len("duke")  
# value of x:  
  
y = len("computer")
```

Built-in Python Function – str() already exists, you use it

- **str()** function
- **Function has:**
 - Name: `str`
 - Input: `an expression`
 - Calculates: `string version of expression's value`
 - Returns back: `string`

Examples:

```
x = str(623)  
# value of x:  
  
y = len( str( 2**8) )  
  
z = str(6 + 8.3)
```

Other Python built-in functions

- **type(something)**
- **int(7.8)**
- **float(4)**
- Returns type of variable `something`
- Returns integer value of decimal number, e.g. 7
- Returns float value of integer, e.g. 4.0

print() function

- **General function has:**
 - Name
 - Maybe inputs
 - Processes or calculates something
 - Has a result
- `print("hi cat")`
 - Name:
 - Input:

Example with lines numbered:

```
→ 1 x = float(6)
   2 print("x is", x)
   3 y = print("x is", x)
   4 print("y is", y)
```

Output:

Writing your own Python function

- **Format:**

```
def nameOfFunction(parameters):
    <body, or lines of code>
    return value # optional, but likely
```
- **Example define function:**

```
def inchesToCentimeters(inches):
    centi = inches * 2.54
    return centi
```
- **Use or call function:**

```
answer = inchesToCentimeters(10.0)
print(answer)
```

Writing your own Python function

- **Parameter**
 - Variable, place holder for a value
 - In parenthesis in first line of definition of function
- **Argument**
 - Expression or value
 - In parenthesis when calling or using a function
- **Example:**

```
def inchesToCentimeters(inches):
    centi = inches * 2.54
    return centi
```
- **Use or call function:**

```
answer = inchesToCentimeters(10.0)
print(answer)
```

What happens when executes?

```
8 def inchesToCentimeters(inches):
9     centi = inches * 2.54
10    return centi
11
12
13 if __name__ == '__main__':
14     answer = inchesToCentimeters(10.0)
15     print(answer)
16     answer = inchesToCentimeters(3.0)
17     print(answer)
```

Output:

Start on line 1 of the file and move line by line
The first 7 lines are blank or are a comment, ignore.

9/6/22

Compsci 101, Fall 2022

32

Let's go see this in Pycharm and add a function

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

newWord = pluralize("fish")
print(newWord)
word1 = "dress"
word2 = pluralize(word1)
print(word2)
word1 = "book"
print(pluralize(word1))
```

Add this function

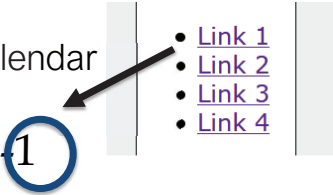
Add these lines of code that call the function

9/6/22

Compsci 101, Fall 2022

47

WOTO – Working Together (breakout groups)

- Given a bitly link
 - Type it in OR click on it on the calendar page
 - <http://bit.ly/101f22-0906-1>
 - What you should do:
 - Introduce yourselves
 - Each person fills out google form
 - Put in your name, email and netid
 - Discuss each question and fill out
 - Be mindful of time
- 

9/6/22

Compsci 101, Fall 2022

48

WOTO: Calling Functions <http://bit.ly/101f22-0906-1>

9/6/22

Compsci 101, Fall 2022

49

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
- Where do you start with problem solving?

The Seven Steps Programming Process: High-level

Steps 1-4:
Devise Algorithm

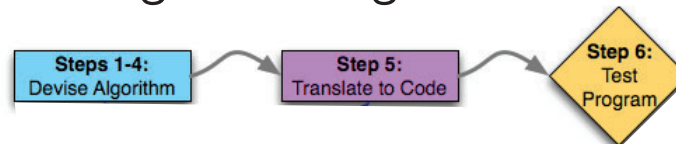
- First part: devise the algorithm
 - The meta-problem solving piece
 - Big/complex enough to be 4 steps (more shortly)

The Seven Steps Programming Process: High-level



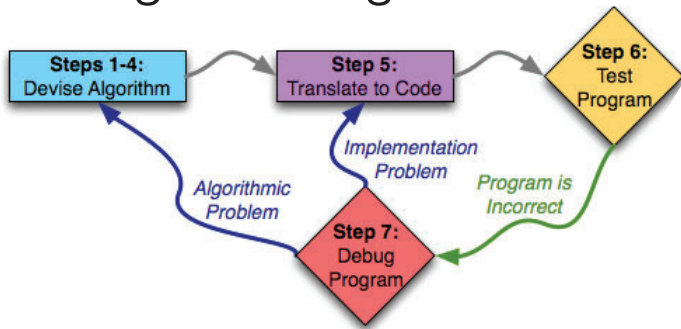
- After devising the algorithm, translate to code
 - Plan first, then code
 - Bridge analogy: blue prints, then construction
 - Essay analogy: outline, then prose

The Seven Steps Programming Process: High-level



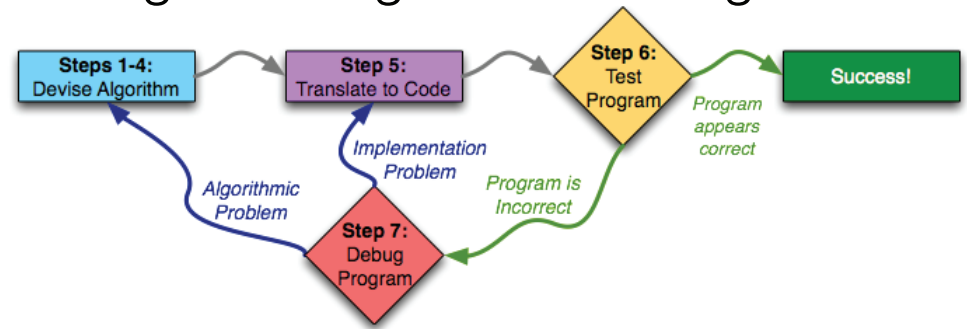
- Next test our program
 - Testing important, often under-taught skill

The Seven Steps Programming Process: High-level



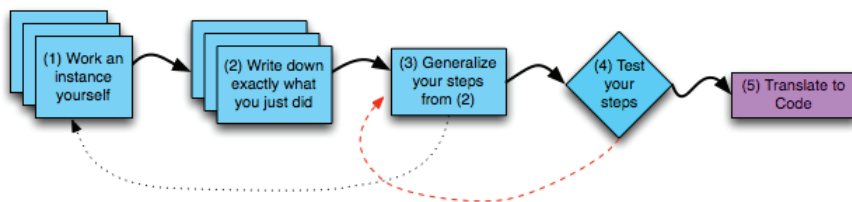
- Ideally would be correct first time; may need to debug
 - Identify problem (with science!)
 - Return to appropriate prior step to fix the problem

The Seven Steps Programming Process: High-level



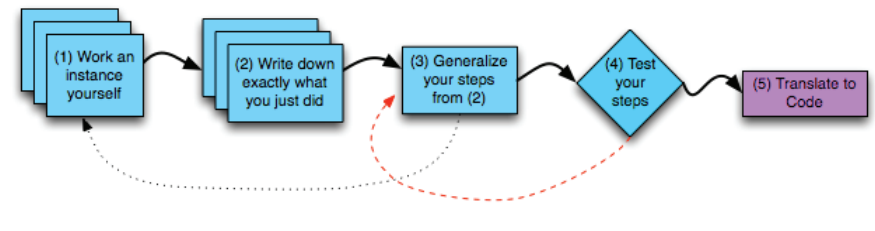
- Work through cycle until program works

Steps 1—4: Devise Algorithm



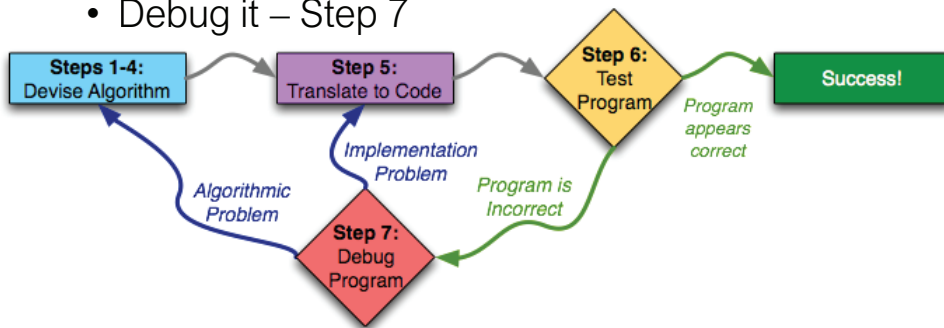
- Steps 1—4: devise the algorithm
 - Learn to do this well, be an excellent programmer
 - Language: does not matter

Steps 1—4: Example: Calculate the average of two numbers



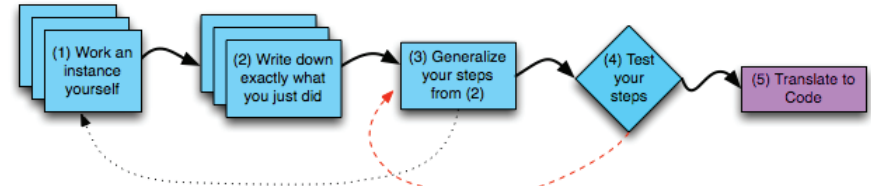
Step 5: let's convert it to code!

- Go to Pycharm
- We will also:
 - Test it – Step 6
 - Debug it – Step 7

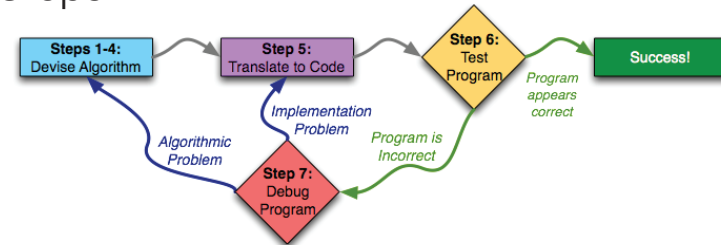


Seven Steps

Steps 1-4



Steps 1-7



Solving Laundry APT

- Navigate to APTs in class website and ...

The screenshot shows the 'CompSci 101, Fall 2022 APTs' page. It includes a navigation menu with 'Home', 'About', 'Dates', 'Labs', 'Assign', 'APTs', 'Help', 'Forms', 'Resources', and 'Sakai'. The main content area is titled 'APT Quiz' and 'APTs'. It provides instructions on how to take the quiz and solve APTs. A table at the bottom lists the APTs and their due dates:

APT	Due Date
APT-1	Sept. 15

Solving Laundry APT

APT Grading: CompSci 101, Fall 2022

This is the webpage for *grading and submitting* your APTs.

Check Grades

[check submissions](#)

Problem Set 1	Details
APT-1, Due on September 15, Complete all six of them	
<input type="radio"/> IntroAPT <input type="radio"/> Bogsquare <input type="radio"/> Cone <input type="radio"/> Grayscale <input type="radio"/> Laundry <input type="radio"/> Gravity	Do first, explains apts in Lecture on 9/6 in Lab 1 on 9/9
Test file: <input type="text" value="Browse..."/> No file selected.	

Solving Laundry APT

- Navigate to APTs in class website and ...

Problem Statement

Consider the problem of trying to do a number of loads of laundry, given only one washer and one dryer. Washing a load takes 25 minutes, drying a load takes 25 minutes, and folding the clothes in a load takes 10 minutes, for a total of 1 hour per load (assuming that the time to transfer a load is built into the timings given). 10 loads of laundry can be done in 10 hours, 600 minutes, using the method of completing one load before starting the next one. Though it can be done faster, see examples.

Write the method, `minutesNeeded`, that returns the shortest time needed to do `m` loads of laundry. In other words, given an integer value representing the number of loads to complete, `m`, determine the smallest number of minutes needed to complete all loads of laundry.

9/6/22

Compsci 101, Fall 2022

66

Specification

```
filename: Laundry.py

def minutesNeeded(m):
    """
    Return integer number of minutes to launder m (integer) loads
    """
    # you write code here
```

Solving Laundry APT – Step 1

WOTO: <http://bit.ly/101f22-0906-2>

- What is important info?

Problem Statement

Consider the problem of trying to do a number of loads of laundry, given only one washer and one dryer. Washing a load takes 25 minutes, drying a load takes 25 minutes, and folding the clothes in a load takes 10 minutes, for a total of 1 hour per load (assuming that the time to transfer a load is built into the timings given). 10 loads of laundry can be done in 10 hours, 600 minutes, using the method of completing one load before starting the next one. Though it can be done faster, see examples.

Write the method, `minutesNeeded`, that returns the shortest time needed to do `m` loads of laundry. In other words, given an integer value representing the number of loads to complete, `m`, determine the smallest number of minutes needed to complete all loads of laundry.

Specification

```
filename: Laundry.py

def minutesNeeded(m):
    """
    Return integer number of minutes to launder m (integer) loads
    """
    # you write code here
```

Solving Laundry APT – Steps 3 and 4

WOTO: <http://bit.ly/101f22-0906-3>

- What is important info?

Problem Statement

Consider the problem of trying to do a number of loads of laundry, given only one washer and one dryer. Washing a load takes 25 minutes, drying a load takes 25 minutes, and folding the clothes in a load takes 10 minutes, for a total of 1 hour per load (assuming that the time to transfer a load is built into the timings given). 10 loads of laundry can be done in 10 hours, 600 minutes, using the method of completing one load before starting the next one. Though it can be done faster, see examples.

Write the method, `minutesNeeded`, that returns the shortest time needed to do `m` loads of laundry.

Specification

```
filename: Laundry.py

def minutesNeeded(m):
    """
    Return integer number of minutes to launder m (integer) loads
    """
    # you write code here
```

Solving an APT

- Create new project
 - File > New Project
 - Existing interpreter (first project you made from installation)
- Create new Python File
 - Right click on project > New > Python File
- Create function within module
 - Name it properly!

9/6/22

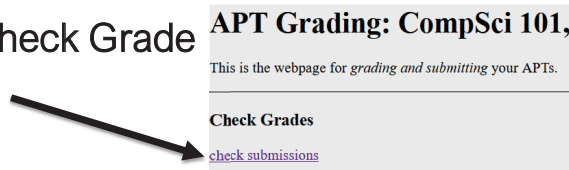
Compsci 101, Fall 2022

74

APT Testing and Submission

- You wrote the code, how is it tested?
 - Submit .py file with function to server
 - Server imports it
 - Server tests and checks by calling your function
- The APT testing framework calls your code!
 - Don't call us, we'll call you: *Hollywood principle*

- Test + Submit + Check Grade



Laundry dissected

```
def minutesNeeded(m):
```

- Wrote formula using code to define a function

Where to put/use what in Python file

- Top: docstring with date and username
- Function definitions right after docstring
- Test code inside `if __name__ == '__main__':`
- Variables inside vs outside a function
 - *Only* use the variables inside that function
 - Therefore, *do not* use the variables outside the function (like in the main)
 - Your code will not work on the server