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

Announcements

- Lab 01 Friday,
  - Complete Prelab before going to lab
- APT-1 out today, due Thursday, January 20
- Assignment 0 due Tuesday, January 18
- 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 Jan 20
- Read Ed Discussion Every Day – You will learn things!
- Reminder: Ed Discussion back channel

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.
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!

---

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: 25
  - Returns back: the actual web page

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")`
      # value of y:

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:
Other Python built-in functions

- **type(something)**
  - Returns type of variable something

- **int(7.8)**
  - Returns integer value of decimal number, e.g. 7

- **float(4)**
  - 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: print
  - Input: “hi cat”
  - “processes”, generates output
  - No return value, returns None

**Example with lines numbered:**

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

Output:

```
x is 6.0
```

**Writing your own Python function**

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

- **Example function:**
  ```python
def inchesToCentimeters(inches):
    centi = inches * 2.54
    return centi
  ```

- **Use or call function:**
  ```python
  answer = inchesToCentimeters(10.0)
  print(answer)
  ```

Output:

```
25.4
```
Writing your own Python function

- Format:
  ```python
def nameOfFunction(parameters):
    <body, or lines of code>
    return value  # optional, but likely
  ```

- Example define function:
  ```python
def inchesToCentimeters(inches):
    centi = inches * 2.54
    return centi
  ```

- Use or call function:
  ```python
  answer = inchesToCentimeters(10.0)
  print(answer)
  ```
  Output: 25.4

What happens when executes?

- 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:
  ```python
def inchesToCentimeters(inches):
    centi = inches * 2.54
    return centi
  ```
  ```python
  answer = inchesToCentimeters(10.0)
  print(answer)
  ```
  Output: 25.4
Let’s go see this in Pycharm and add a function

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

Add this function

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

Add these lines of code that call the function

WOTO – Working Together (breakout groups)

- Given a bitly link
  - Type it in OR click on it on the calendar page

- 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

WOTO: Calling Functions

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

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

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

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 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

Seven Steps

Steps 1-4

- (1) Work an instance yourself
- (2) Write down exactly what you just did
- (3) Generalize your steps from (2)
- (4) Test your steps
- (5) Translate to code

Steps 1-7

- (1) Work an instance yourself
- (2) Write down exactly what you just did
- (3) Generalize your steps from (2)
- (4) Test your steps
- (5) Translate to code

Solving Laundry APT

- Navigate to APTs in class website and ...

CompSci 101, Spring 2022

APT Quiz

Start the APT quiz on Sakai under quizzes, but not until you are ready to take the quiz.

APT Quizzes

See below for hints on what to do if your APT doesn’t run.

For each problem in an APT set, complete these steps by the due date
- First click on the APT set below to go to the APT page.
  - Write the code, upload the file and click the Submit link.
  - Check your grade on the grade code page by clicking on check submissions.

In solving APTs, your program should work for all cases, not just the test cases we provide. We may test your program on additional data.

APT: Due Date

- APT: January 20
Solving Laundry APT

**APT Grading: CompSci 101, Spring 2022**

This is the webpage for grading and submitting your APTs.

**Check Grades**

- check submissions

---

**Problem Set 1**

| APT-1, Due on Jan 20, Complete all six of them |
|-----------------|-------------------|
| ○ IntroAPT | Do first, explains apts |
| ○ Gravity | in Lab 1 on 1/14 |
| ○ Bobsquare | |
| ○ Cone | |
| ○ Grayscale | in Lecture 1/13 |
| ○ Laundry | |

**Test file: Browse... No file selected.**

---

1/13/22 Compsci 101, Spring 2022 63

---

**Solving Laundry APT – Step 1**


- 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 a loads of laundry. In other words, given an integer value representing the number of loads to complete, n, determine the smallest number of minutes needed to complete all loads of laundry.

---

**Reading an APT**

- **Step 1: Work an instance yourself**
- **Step 2: Write down exactly what you just did**
- **Step 3: Generalize your steps**
- **Step 4: Test your steps (with new input)**

**Examples**

1. \( m = 1 \)
   - returns: 60
   - You must was minutes.
2. \( m = 2 \)
   - returns: 85

---

**Specfication**

```python
filename: Laundry.py
def minutesNeeded(m):
    # you write code here
```

---

**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 a loads of laundry. In other words, given an integer value representing the number of loads to complete, \( n \), determine the smallest number of minutes needed to complete all loads of laundry.
Solving Laundry APT – Steps 3 and 4


- 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 n loads.

**Specification**

```python
filename: Laundry.py
def minutesNeeded(m):
    ***
    Return integer number
    ***
    # 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!

Names and Return 0 Submission

- Take small steps to get all green!

APT Correct ➔ The Green Dance(Fall 2020)
APR 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

Testing Laundry – three ways

1) Run on the apt page
   - Need internet connection, may take time

2) Locally in Python Program Laundry
   - Get it working before you use apt page

3) Run in Python Console
   - Must import Laundry
   - Must specify which file minutesNeeded is in
   
   ```python
   >>> minutesNeeded(2)
   85
   ```

```python
if __name__ == '__main__':
    num = 1
    print("m is", num, minutesNeeded(num))
    num = 2
    print("m is", num, minutesNeeded(num))
    num = 3
    print("m is", num, minutesNeeded(num))
    num = 10
    print("m is", num, minutesNeeded(num))
```
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