

EEE110 Computer Programming

Functions

Dr Kasim Zor

March 21, 2023

Functions



Question 1. Defining and Calling Functions - Part 1/3 (15 minutes)

Professional Appliance Service, Inc. offers maintenance and repair services for household appliances. The owner wants to give each of the company's service technicians a small handheld computer that displays step-by-step instructions for many of the repairs that they perform. To see how this might work, the owner has asked you to develop a program that displays the following instructions for disassembling an Acme laundry dryer:

- Step 1: Unplug the dryer and move it away from the wall.
- Step 2: Remove the six screws from the back of the dryer.
- Step 3: Remove the dryer's back panel.
- Step 4: Pull the top of the dryer straight up.

During your interview with the owner, you determine that the program should display the steps one at a time. You decide that after each step is displayed, the user will be asked to press the Enter key to see the next step.

Functions



Question 1. Defining and Calling Functions - Part 2/3 (15 minutes)

Here is the algorithm in pseudocode:

- Display a starting message, explaining what the program does.
- Ask the user to press Enter to see step 1.
- Display the instructions for step 1.
- Ask the user to press Enter to see the next step.
- Display the instructions for step 2.
- Ask the user to press Enter to see the next step.
- Display the instructions for step 3.
- Ask the user to press Enter to see the next step.
- Display the instructions for step 4.

This algorithm lists the top level of tasks that the program needs to perform and becomes the basis of the program's main function.

Functions



Question 1. Defining and Calling Functions - Part 3/3 (15 minutes)



As you can see from the hierarchy chart, the main function will call several other functions. Here are summaries of those functions:

- startup message: This function will display the starting message that tells the technician what the program does.
- step1: This function will display the instructions for step 1.
- step2: This function will display the instructions for step 2.
- step3: This function will display the instructions for step 3.
- step4: This function will display the instructions for step 4.

Between calls to these functions, the main function will instruct the user to press a key to see the next step in the instructions.

Solution 1 - Part 1/3

```
def main():
    startup_message()
    #input('Press Enter to see Step 1.')
    step1()
    #input('Press Enter to see Step 2.')
    step2()
    #input('Press Enter to see Step 3.')
    step3()
    #input('Press Enter to see Step 4.')
    step4()
def startup_message():
    print('This program tells you how to')
    print('disassemble an ACME laundry dryer.')
    print('There are 4 steps in the process.')
```

Solution 1 - Part 2/3

```
def step1():
    print('Step 1: Unplug the dryer and')
    print('move it away from the wall.')
def step2():
    print('Step 2: Remove the six screws')
    print('from the back of the dryer.')
def step3():
    print('Step 3: Remove the back panel')
    print('from the dryer.')
def step4():
    print('Step 4: Pull the top of the')
    print('dryer straight up.')
```

Solution 1 - Part 3/3

```
main()
## This program tells you how to
## disassemble an ACME laundry dryer.
## There are 4 steps in the process.
## Step 1: Unplug the dryer and
## move it away from the wall.
## Step 2: Remove the six screws
## from the back of the dryer.
## Step 3: Remove the back panel
## from the dryer.
## Step 4: Pull the top of the
## dryer straight up.
```

Question 2. Passing an Argument to a Function - Part 1/2 (10 minutes)

Your friend Michael runs a catering company. Some of the ingredients that his recipes require are measured in cups. When he goes to the grocery store to buy those ingredients, however, they are sold only by the fluid ounce. He has asked you to write a simple program that converts cups to fluid ounces. You design the following algorithm:

- 1 Display an introductory screen that explains what the program does.
- 2 Get the number of cups.
- 3 Convert the number of cups to fluid ounces and display the result.

This algorithm lists the top level of tasks that the program needs to perform and becomes the basis of the program's main function.

Question 2. Passing an Argument to a Function - Part 2/2 (10 minutes)

The main function will call two other functions. Here are summaries of those functions:

- **intro:** This function will display a message on the screen that explains what the program does. Example: This program converts measurements in cups to fluid ounces. For your reference the formula is: 1 cup = 8 fluid ounces
- **cups to ounces:** This function will accept the number of cups as an argument and calculate and display the equivalent number of fluid ounces.

In addition to calling these functions, the main function will ask the user to enter the number of cups. This value will be passed to the `cups_to_ounces` function.

Solution 2 - Part 1/2

```
def main():
    intro()
    cups_needed = 3 #int(input('Enter the number of cups: '))
    cups_to_ounces(cups_needed)
def intro():
    print('This program converts measurements')
    print('in cups to fluid ounces. For your')
    print('reference the formula is:')
    print('    1 cup = 8 fluid ounces')
def cups_to_ounces(cups):
    ounces = cups * 8
    print('That converts to', ounces, 'ounces.')
```

Solution 2 - Part 2/2

```
main()
## This program converts measurements
## in cups to fluid ounces. For your
## reference the formula is:
##    1 cup = 8 fluid ounces
## That converts to 24 ounces.
```

Question 3. Using Global Constants (15 minutes)

Marilyn works for Integrated Systems, Inc., a software company that has a reputation for providing excellent fringe benefits. One of their benefits is a quarterly bonus that is paid to all employees. Another benefit is a retirement plan for each employee. The company contributes 5 percent of each employee's gross pay and bonuses to their retirement plans. Marilyn wants to write a program that will calculate the company's contribution to an employee's retirement account for a year. She wants the program to show the amount of contribution for the employee's gross pay and for the bonuses separately. Here is an algorithm for the program:

- Get the employee's annual gross pay.
- Get the amount of bonuses paid to the employee.
- Calculate and display the contribution for the gross pay.
- Calculate and display the contribution for the bonuses.

Solution 3 - Part 1/2

```
CONTRIBUTION_RATE = 0.05
def main():
    gross_pay = 80000.0 #float(input('...'))
    bonus = 20000.0 #float(input('...'))
    show_pay_contrib(gross_pay)
    show_bonus_contrib(bonus)
def show_pay_contrib(gross):
    contrib = gross * CONTRIBUTION_RATE
    print('Contribution for gross pay: $', \
          format(contrib, ',.2f'), \
          sep='')
```

Solution 3 - Part 2/2

```
def show_bonus_contrib(bonus):
    contrib = bonus * CONTRIBUTION_RATE
    print('Contribution for bonuses: $', \
          format(contrib, ',.2f'), \
          sep='')
main()
## Contribution for gross pay: $4,000.00
## Contribution for bonuses: $1,000.00
```

Question 4. Using Random Numbers (10 minutes)

Dr. Kimura teaches an introductory statistics class and has asked you to write a program that he can use in class to simulate the rolling of dice. The program should randomly generate two numbers in the range of 1 through 6 and display them. In your interview with Dr. Kimura, you learn that he would like to use the program to simulate several rolls of the dice, one after the other. Here is the pseudocode for the program:

- While the user wants to roll the dice:
 - Display a random number in the range of 1 through 6
 - Display another random number in the range of 1 through 6
 - Ask the user if he or she wants to roll the dice again

You will write a while loop that simulates one roll of the dice and then asks the user if another roll should be performed. As long as the user answers “y” for yes, the loop will repeat.

Solution 4

```
import random
MIN = 1
MAX = 6
def main():
    again = 'y'
    while again == 'y' or again == 'Y':
        print('Rolling the dice... Their values are:')
        print(random.randint(MIN, MAX))
        print(random.randint(MIN, MAX))
        again = 'Q' #input('...')
main()
## Rolling the dice... Their values are:
## 4
## 4
```



Question 5. Random for Representing Other Values (10 minutes)

Dr. Kimura was so happy with the dice rolling simulator that you wrote for him, he has asked you to write one more program. He would like a program that he can use to simulate ten coin tosses, one after the other. Each time the program simulates a coin toss, it should randomly display either “Heads” or “Tails”. You decide that you can simulate the tossing of a coin by randomly generating a number in the range of 1 through 2. You will write an if statement that displays “Heads” if the random number is 1, or “Tails” otherwise. Here is the pseudocode:

- Repeat 10 times:
 - If a random number in the range of 1 through 2 equals 1 then:
 - Display ‘Heads’
 - Else:
 - Display ‘Tails’

Because the program should simulate 10 tosses of a coin you decide to use a for loop.



Solution 5 - Part 1/2

```
import random
HEADS = 1
TAILS = 2
TOSSES = 10
def main():
    for toss in range(TOSSES):
        if random.randint(HEADS, TAILS) == HEADS:
            print('Heads')
        else:
            print('Tails')
```



Solution 5 - Part 2/2

```
main()
## Tails
## Heads
## Tails
## Heads
## Heads
## Heads
## Tails
## Tails
## Heads
## Tails
```



Question 6. Modularizing with Functions - Part 1/3 (20 minutes)

Hal owns a business named Make Your Own Music, which sells guitars, drums, banjos, synthesizers, and many other musical instruments. Hal’s sales staff works strictly on commission. At the end of the month, each salesperson’s commission is calculated according to Table:

Sales This Month	Commission Rate
Less than \$10,000	10%
\$10,000-\$14,999	12%
\$15,000-\$17,999	14%
\$18,000-\$21,999	16%
\$22,000- or more	18%

For example, a salesperson with \$16,000 in monthly sales will earn a 14 percent commission (\$2,240). Another salesperson with \$18,000 in monthly sales will earn a 16 percent commission (\$2,880). A person with \$30,000 in sales will earn an 18 percent commission (\$5,400).

Question 6. Modularizing with Functions - Part 2/3 (20 minutes)

Because the staff gets paid once per month, Hal allows each employee to take up to \$2,000 per month in advance. When sales commissions are calculated, the amount of each employee's advanced pay is subtracted from the commission. If any salesperson's commissions are less than the amount of their advance, they must reimburse Hal for the difference. To calculate a salesperson's monthly pay, Hal uses the following formula:

$$\text{pay} = \text{sales} \times \text{commission rate} - \text{advanced pay}$$

Question 6. Modularizing with Functions - Part 3/3 (20 minutes)

Hal has asked you to write a program that makes this calculation for him. The following general algorithm outlines the steps the program must take.

- 1 Get the salesperson's monthly sales.
- 2 Get the amount of advanced pay.
- 3 Use the amount of monthly sales to determine the commission rate.
- 4 Calculate the salesperson's pay using the formula previously shown. If the amount is negative, indicate that the salesperson must reimburse the company.

Solution 6 - Part 1/4

```
def main():
    sales = get_sales()
    advanced_pay = get_advanced_pay()
    comm_rate = determine_comm_rate(sales)

    pay = sales * comm_rate - advanced_pay
    print('The pay is $', format(pay, ',.2f'), sep='')

    if pay < 0:
        print('The salesperson must reimburse')
        print('the company.')
```

Solution 6 - Part 2/4

```
def get_sales():
    monthly_sales = 14650.00 #float(input('Enter the monthly sales: '))
    return monthly_sales

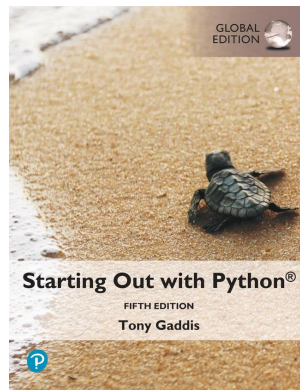
def get_advanced_pay():
    print('Enter the amount of advanced pay, or')
    print('enter 0 if no advanced pay was given.')
    advanced = 1000.0 #float(input('Advanced pay: '))
    return advanced
```

Solution 6 - Part 3/4

```
def determine_comm_rate(sales):
    if sales < 10000.00:
        rate = 0.10
    elif sales >= 10000 and sales <= 14999.99:
        rate = 0.12
    elif sales >= 15000 and sales <= 17999.99:
        rate = 0.14
    elif sales >= 18000 and sales <= 21999.99:
        rate = 0.16
    else:
        rate = 0.18
    return rate
```

Solution 6 - Part 4/4

```
main()
## Enter the amount of advanced pay, or
## enter 0 if no advanced pay was given.
## The pay is $758.00
```



Aforementioned contents are adapted from the book:

- 'Starting out with Python' written by Tony Gaddis.