COSC 1336 Homework 2

Relevant reading: Chapter 3

Due: Sep. 20, 2:30 pm

(Late date: Sep. 27, 2:30 pm)

50 Points

Problem 1. [**8 points**]

a. The first line of a function definition is known as the function ________.
b. The rest of the lines in a function definition after the first line are known together as the function ________.
c. You ________ a function to execute it.
d. A special variable that receives a piece of data when a function is called is a(n) ________.
e. A(n) ________ piece of data that is sent into a function when it is called.
f. A(n) ________ is a variable that is created inside a function.
g. The parts of a program where a variable may be referenced are known collectively as the ________ of the variable.
h. A variable that can be referenced in every function in a program is a(n) ________.

Problem 2. [**4 points**]

a. Write a function named `times_ten`. The function should accept an argument and display the product of its argument multiplied by 10.
b. Examine the following function header, and then write a statement that calls the function, passing 12 as an argument.

```
def show_value(quantity):
```

c. Look at the following function header:

```
def my_function(a, b, c):
```

Now look at the following call to the `my_function`:

```
my_function(3, 2, 1):
```

When this call executes, what value will be assigned to `a`, `b` and `c`?

d. Given the same function header for `my_function`, what values will be assigned to `a`, `b` and `c` when the following function call executes?

```
my_function(5, c = 4, b = 6)
```

Problem 3. [**20 points**] For the code in each subproblem below, do two things:

i. Draw a stack diagram for the indicated instruction in the program (4 points).
ii. Say what the program's output will be when execution has finished (1 point). Obviously you could answer this part just by running the code in the interpreter, but you should be sure that the answer you give matches with your diagram.

Stack diagrams are not covered in the textbook, but they are a useful way to trace the execution of a program that contains functions. Examples of stack diagrams can be seen in the Powerpoint slides posted on Blackboard in the Resources section under "Code from lecture".

a. [**5 points**]

```
def print_times_ten(x):
    print(x * 10)  # <=== draw stack diagram when execution is here

def print_diff_times_ten(a, b):
    diff = a - b
    print_times_ten(diff)

z = 14
y = 10
print_diff_times_ten(z, y)
```

b. [**5 points**]

```
def print_difference(x, y):
    diff = x - y
    print(diff)  # <=== draw stack diagram when execution is here

def main():
    x = 4
    y = 8
    print_difference(y, x) # careful! note order of arguments

main()
```

c. [**5 points**]

```
def main():
    x = 6
    y = x * 2
    change_us(x, y)
    print(x, y)

def change_us(a, b):
    a = 0
    b = 0  # <== draw stack diagram here

main()
```

d. [**5 points**] For this one, assume the user enters `10` for the number of tickets, and `12.50` for the ticket price.

```
def print_money(dollars):
    print("$", format(dollars, ".2f"), sep="") # <=== draw stack diagram

def display_total_due(num_tickets, ticket_price):
    total_due = num_tickets * ticket_price
    print("Total due: ")
    print_money(total_due)

tickets = int(input("Enter the number of tickets: "))
price = float(input("Enter the price of each ticket: $"))
display_total_due(tickets, price)
```

Problem 4. [**6 points**] Look at the following function definition:

```
def my_function(a, b, c):
    d = (a + c) / b
    print(d)
```

a. [**3 points**] Write a statement that calls this function and uses keyword arguments to pass 2 into `a`, 4 into `b`, and 6 into `c`.

b. [**3 points**] What value will be displayed when the function call executes?

Problem 5. [**12 points**] Each of the following pieces of code has a bug. Describe the bug and say how you'd fix it. Your answers should be complete sentences, but only one or two sentences should suffice.

a. [**4 points**]

```
def print_twice():
    print(message)
    print(message)

message = input("Enter a message to display twice: ")
print_twice()
```

b. [**4 points**]

```
def print_course_name():
    print("COSC 1336")

print_course_name
```

c. [**4 points**]

```
def print_average(a, b, c):
    ave = (a + b + c) / 2
    print("Average is: ", ave)

x = 4
y = 5
print_average(x, y)
```