Master of Engineering in Internetworking

Lab # 1

INWK 6312

Question 1

Start the Python interpreter and use it as a calculator. Python's syntax for math operations is almost the same as standard mathematical notation. For example, the symbols `+`, `-` *and* `/` *denote addition, subtraction and division, as you would expect. The symbol for multiplication is* `*`.

If you run a 10 kilometer race in 43 minutes 30 seconds, what is your average time per mile? What is your average speed in miles per hour? (Hint: there are 1.61 kilometers in a mile).

Question 2

Assume that we execute the following assignment statements:

```
width = 17
height = 12.0
delimiter = '.'
```

For each of the following expressions, write the value of the expression and the type (of the value of the expression).

1. `width/2`
2. `width/2.0`
3. `height/3`
4. `1 + 2 * 5`
5. `delimiter * 5`

Question 3

Practice using the Python interpreter as a calculator:

1. *The volume of a sphere with radius r is 4/3 πr^3. What is the volume of a sphere with radius 5? Hint: 392.7 is wrong!*
2. *Suppose the cover price of a book is $24.95, but bookstores get a 40% discount. Shipping costs $3 for the first copy and 75 cents for each additional copy. What is the total wholesale cost for 60 copies?*
3. *If I leave my house at 6:52 am and run 1 mile at an easy pace (8:15 per mile), then 3 miles at tempo (7:12 per mile) and 1 mile at easy pace again, what time do I get home for breakfast?*

Question 4

Python provides a built-in function called `len` *that returns the length of a string, so the value of* `len('allen')` *is 5.*

Write a function named `right_justify` *that takes a string named* `s` *as a parameter and prints the string with enough leading spaces so that the last letter of the string is in column 70 of the display.*

```
>>> right_justify('allen')                                        allen
```

Question 5

A function object is a value you can assign to a variable or pass as an argument. For example, `do_twice` *is a function that takes a function object as an argument and calls it twice:*

```
def do_twice(f):
    f()
    f()
```

Here's an example that uses `do_twice` *to call a function named* `print_spam` *twice.*

```
def print_spam():
    print 'spam'

do_twice(print_spam)
```

1. *Type this example into a script and test it.*
2. *Modify* `do_twice` *so that it takes two arguments, a function object and a value, and calls the function twice, passing the value as an argument.*
3. *Write a more general version of* `print_spam`*, called* `print_twice`*, that takes a string as a parameter and prints it twice.*
4. *Use the modified version of* `do_twice` *to call* `print_twice` *twice, passing* `'spam'` *as an argument.*
5. *Define a new function called* `do_four` *that takes a function object and a value and calls the function four times, passing the value as a parameter. There should be only two statements in the body of this function, not four.*

Question 6

This exercise can be done using only the statements and other features we have learned so far.

1. *Write a function that draws a grid like the following:*

```
+ - - - - + - - - - +
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
+ - - - - + - - - - +
|         |         |
|         |         |
|         |         |
|         |         |
|         |         |
+ - - - - + - - - - +
```

   Hint: to print more than one value on a line, you can print a comma-separated sequence:

   ```
   print '+', '-'
   ```

   If the sequence ends with a comma, Python leaves the line unfinished, so the value printed next appears on the same line.

   ```
   print '+',
   print '-'
   ```

   The output of these statements is `'+ -'`.

   A `print` *statement all by itself ends the current line and goes to the next line.*

2. *Write a function that draws a similar grid with four rows and four columns.*