Introduction to Programming (CS102) 2006/2007 — Practical 5

Work through the following sections. Seek assistance whenever needed. From http://schmidt.nuigalway.ie/cs102/python files with Python programs can be downloaded. Present your results to one of the demonstrators, so that a record of your achievements can be kept.

13. True or False?

1. In Python ’4’ + ’5’ equals ’45’.
2. Python lists are mutable, but strings are not.
3. ASCII is a standard for representing characters using numeric codes.
4. The split() function breaks a string into a list of substrings, and join() does the opposite.

14. Multiple Choice.

1. What function gives the ASCII value of a character?
   a) ord b) ascii c) chr d) eval
2. Which of the following can not be used to convert a string of digits into a number?
   a) int b) float c) str d) eval
3. Which string library function converts all the characters of a string to upper case?
   a) capitalize b) capwords c) uppercase d) upper

15. Programming Exercises.

1. Given the initial statements

   ```
   import string
   s1 = ’spam’
   s2 = "ni!"
   ```

   show the result of evaluating each of the following string expressions.
   (a) "The Knights who say, " + s2
   (b) 3 * s1 + 2 * s2
   (c) s1[1]
   (d) s1[1:3]
   (e) s1[2] + s2[:2]
   (f) s1 + s2[-1]
   (g) string.upper(s1)
   (h) string.ljust(string.upper(s2), 4) * 3
2. Given the same initial statements as in the previous problem, show a Python expression that could construct each of the following results by performing string operations on s1 and s2.

(a) `'NI'`
(b) `'ni!spamni!'`
(c) `'Spam Ni!  Spam Ni!  Spam Ni!'`
(d) `'spam'`
(e) `['sp', 'm']`
(f) `'spm'`

3. `append` is a *list* function that *appends* another element to an existing list:

   ```
   >>> list = [2, 3, 5]
   >>> list.append(7)
   >>> list
   [2, 3, 5, 7]
   ```

   Because strings are immutable, you cannot append characters to strings. That's why in the `numbers2text.py` program the accumulator `message` has to be updated with a statement like

   ```
   message = message + chr(n)
   ```

   which is somewhat inefficient, since it creates an entirely new string containing a copy of the message on each iteration.

   Rewrite this program in such a way, that the accumulator is a list of characters, which is updated by `append`ing, rather than concatenation, and is at the end of the program turned into a single string by `join`ing the list, using an empty string as separator between the characters.
4. A 5-point quiz is graded on the scale 5:A, 4:B, 3:C, 2:D, 0 or 1:F. Write a program (similar to `month.py`) that accepts a quiz score as input and prints out the corresponding grade.
5. A 100-point exam is graded on the scale 90-100:A, 80-89:B, 70-79:C, 60-69:D, less than 60:F. Write a program that accepts an exam score as input and prints out the corresponding grade. (Hint: use integer division.)
6. An *acronym* is a word formed by taking the first letters of the words in a phrase. For example, RAM is an acronym for "Random Access Memory". Write a program that allows a user to type in a phrase and then outputs the acronym for that phrase. The acronym should be all uppercase, even if the words in the phrase are not capitalized.
7. A *Caesar cipher* is a simple substitution cipher based on the idea of shifting each letter of the plaintext message by a fixed number (called the key) of positions in the alphabet. For example, if the key value is 2 the word `"hello"` would be encoded as `"jgnnq"`. The original message can be recovered by shifting the encoded word backwards.

   Write a program that can encode Caesar ciphers. The input to the program will be a string of text and the value of the key. The output will be an encoded message, where each character `c` in the original message its shift `chr(ord(c) + key)` about `key` positions in the ASCII character set.