University of HUDDERSFIELD

python powered

CFS2160: Programming Stream

Tutorial/Practical 3

Objects and Classes

Introduction

Now we can write more complex programs making use of a range of different data types.

You know by now that activities marked ✠ should be included in your logbook for assessment. And also that activities in *italics* are for those who find the rest straightforward. They are entirely optional.

Purpose

By the end of the week you should:

- Be able to write programs that take input from the user, and store this input in an appropriate data structure.
- Be able to do operations on strings to produce useful results.
- Understand how to use Tuples, and when to use them.

If in doubt - Ask!

Next Week

Next week your tutor will ask to see your up-to-date Log Book. So be sure to bring it with you.

Activities

1. Make your own version of the average student mark program from the lecture (it's on the slides), and run it. What does the `\'` in `student\'s` do on the first line?

2. Type the following at the Python console, and make sure you understand the results. Ask if you don't!

```
>>> b = True
>>> type (b)
>>> b = 'True'
>>> type (b)
>>> 'True' == True
```

3. Suppose you had a variable created at the Python prompt, like so:
   ```
   >>> robin = 'Boldly Run Away!'
   ```
   Can you predict what the following slices of this string will return? Try it out to check the answers.
   ```
   >>> robin [1]
   >>> robin [-1]
   >>> robin [:6]
   >>> robin [7:][:-1]
   >>> robin [::2]
   ```

4. ↯
   Last week you wrote a program that printed a greeting to the user after they entered their name. It worked something like this:
   ```
   Hello, who are you? Arthur
   Hello, Arthur. It is good to meet you.
   ```
   Make a copy of the program, and amend it so that it works like this:
   ```
   Hello, who are you? arthur
   Hello, Arthur. It is good to meet you.
   ```
   That is, it *capitalises* the first letter of the entered name if the user does not, and also converts the rest of the name to lowercase.
   Include a screenshot of the program in your assessment logbook. Show it working when the user enters their name in all UPPERCASE letters.
   Hint: This is easy, and a tiny change, and the answer is on the lecture slides!

5. Of course, everyone who uses the program is a Knight, and should be addressed as "Sir". Take another copy of the program, and amend it so that it works like this:
   ```
   Hello, who are you? robin
   Hello, Sir Robin. It is good to meet you.
   ```

6. *Of course, again, Arthur is not a Knight, but a King. Taking another copy, write a program that works as in the previous activity unless the name is Arthur, in which case a different message should be displayed. Remember that the name could be entered in uppercase, lowercase, or a mix. It should work like this:*
   ```
   Hello, who are you? robin
   Hello, Sir Robin. It is good to meet you.
   Hello, who are you? arthur
   My Liege! It is good to meet you.
   ```

7. *The program for exercise 5 above would fail if the user did enter "Sir " as the first four characters of their name. Can you work out how to fix that one?*

8. Create a tuple containing names at the Python prompt. For example:
   ```
   >>> knights = ('robin', 'galahad', 'bors', 'bedevere',)
   ```
   Write down (and test) slices that will find the first, last, and third entries in the tuple.

9. Write and test a statement that will add a new string `'launcelot'` to the end of the tuple.

10. ↯
    Write a program that prompts the user to enter the prices of five items in the sweet shop. Assume that all items cost less than £1, so the user will enter them as "10p", "78p" and so on. Display the total price, average price, highest price, and lowest price.

It will look something like this when run:

```
Enter the price of the first sweet:  12p
Enter the price of the second sweet: 55p
Enter the price of the third sweet:  65p
Enter the price of the fourth sweet: 72p
Enter the price of the fifth sweet:  10p

Total Price:   214p
Average Price: 42.8p
Highest Price: 72p
Lowest Price:  10p
```

Final Note

You have now generated several different versions of some of your programs. Can you see how Git could be helping here? Go back and see if you are not sure!