Birkbeck
UNIVERSITY OF LONDON

Introduction to Programming

Python Lab 4:
Arithmetic

Ping Brennan (p.brennan@bbk.ac.uk)

22 October 2019 or
7 February 2020

Getting Started

- Create a new folder in your disk space with the name **PythonLab4**

- Launch the Python Integrated Development Environment (IDLE) - begin with the **Start** icon in the lower left corner of the screen.

- If you are in a DCSIS laboratory, search using the keyword **Python** and click on **IDLE (Python 3.6 64-bit)**

  A window with the title **Python 3.6.2** should appear. This window is the *Shell*.

Getting Started (2)

- If you are in the ITS laboratory MAL 109, then right mouse click on the **Start** icon in the lower left corner of the screen.

  A list of menu options should appear and click on ***Search***. Type ***Python*** in the search text box at the bottom of the pop-up window. A list of Apps should appear and select

  Python 3.4 IDLE(PythonGUI)

  A window with the title **Python 3.4.3 Shell** should appear. This window is the *Shell*.

- In the *Shell* click on **File**. A drop down menu will appear.

  Click on **New File.** A window with the `title` **Untitled** should appear. This window is the *Editor*.

Getting Started (3)

- In the *Editor*, click on **File**, and then in the drop down menu click on **Save As...** .

  A window showing a list of folders should appear.
  - To search any folder on the list, double click on the folder.
  - Find the folder PythonLab4 and double click on it.
  - In the box **File name** at the bottom of the window
    1. Type `BookStore.py`
    2. Then click on the button **Save** in the lower right corner of the window.

  The title of the *Editor* should change to show the location of the file `BookStore.py`

Program `BookStore.py` calculates the price of an order for books

- **Question 2: Problem statement**

  The following pseudo code describes how a bookstore computes the price of an order from the total price and the number of books that were ordered (PFE Business P2.32).

  1. Read the total book price and the number of books
  2. Compute the tax (7.5 per cent of the total book price), that is,

```
tax  = total book price  *  (7.5/100)
```

  3. Compute the shipping charge ($2 per book), that is,

```
shipping charge  =  number of books  *  2.00
```

  4. The price of the order is the sum of the total book price, the tax and the shipping charge, that is,

```
order price =  total book price  + tax + shipping charge
```

  5. Print the price of the order.

Program `BookStore.py` calculates the price of an order for books (2)

- **Problem solving** - Convert the pseudo code on page 5 into a Python program.
  - Prices are in dollars and cents, e.g. $3.67.
  - The total book price and the number of books can be assigned values in the program.
  - For example,

```
totalBookPrice = 3.67
```

Program `BookStore.py` calculates the price of an order for books (3)

- **Problem solving (continued)**
  - Alternatively, the total book price and the number of books can be entered from the keyboard using the function `input` in the following style,

```
totalBookPrice = float(input("Enter the total book price: "))
```

  - If the total book price is $3.67 then the characters `3.67` are typed on the keyboard, after which the `Enter` key is pressed.
    The function `input` returns a string `"3.67"` to the calling program.
    The function `float` converts the string to a floating point number, in this case `3.67`
    Thus,

totalBookPrice = 3.67

Program `BookStore.py` calculates the price of an order for books (4)

- Below is a skeleton program to help you get started:

```
# Read the total book price
totalBookPrice = float(input("Enter the total book price: "))

# Use the function input to read keyboard input for the
# number of books, and then use the function int to
# convert the string to an integer.
numberOfBooks = int(input( To Do: add a suitable prompt here ))

# Compute the tax (7.5 per cent of the total book price)

tax = totalBookPrice * (7.5/100)

# Calculate the shipping charge ($2 per book)
 Add a Python statement here to calculate the shipping charge
# Calculate the price of the order which is the sum of
# the total book price, the tax and the shipping charge.
 Add a Python statement here to calculate the price of the order
# Print the price of the order.
 Call the function print to display the price of the order
```

Program `BookStore.py` calculates the price of an order for books (5)

- Provide a comment at the beginning of the program to explain the purpose of the program, along with your name and the date.
- Save the program to the file `BookStore.py`
- Run your program.
- Note if the value of the variable is significant and does not change, then use only uppercase letters and underscores in the name. For example,

```
TAX_RATE = 7.5/100
tax = totalBookPrice * TAX_RATE
```

Program `DollarsAndCents.py` extracts the dollars and cents from a price

- Create a new Editor for a new file called `DollarsAndCents.py`
- **Question 3: Problem statement**

  The following pseudo code describes how to extract the dollars and cents from a price given as a floating point value.

  For example, a price of 2.95 yields the values `2` and `95` for the dollars and cents (PFE Business P2.34).

  1. Read the `price`
  2. Convert the `price` to an integer using the function `int` and store the integer in the variable `dollars`.
  3. Multiply the difference, `price` – `dollars` by `100` and add `0.5`.
  4. Convert the result to an integer using the function `int` and store the integer in the variable `cents`.
  5. Print the `dollars` and `cents`.

Program `DollarsAndCents.py` extracts the dollars and cents from a price (2)

- **Problem Solving** - below is a skeleton program to help you get started:

```
# Read in the price using the function input and then use the
# function float to convert the string to a floating-point value.
price = float(input("Enter a price: "))

# Convert the price to an integer using the function int
# and store the integer in the variable dollars
dollars = int(price)
# Create a variable CENTS_IN_A_DOLLAR and assign it the value 100
Add a Python statement to create and initialise the variable
CENTS_IN_A_DOLLAR
# Multiply the difference(price - dollars) by CENTS_IN_A_DOLLAR
# and add 0.5; and then store the answer in a new variable result
Add a Python statement to do the calculation as described in above comment
# Convert the 'result' to an integer using the function int
# and store the integer in the variable cents.
Add a Python statement to do the conversion as described in above comment
# Print the dollars and cents.
Call the function print to display the dollars and cents
```

Program `DollarsAndCents.py` extracts the dollars and cents from a price (3)

- Provide a comment at the beginning of the program to explain the purpose of the program, along with your name and the date.

- Save your program to the file `DollarsAndCents.py`

- Run your program.

Note: The function `int` truncates. For example, the function call `int(2.9)` returns the integer `2`.

Think about: What happens if a negative price is input?

Program `Conversion.py` converts a measurement in meters to miles, feet and inches

- Create a new Editor for a new file called `Conversion.py`

- **Question 4: Problem statement**

  Write a program that prompts the user for a measurement in meters and then converts it to miles, feet and inches (PFE P2.6). The following data is provided.

  1 mile = 1609.34 meters

  1 mile = 1760 yards

  1 yard = 3 feet

  1 foot = 12 inches

Program `Conversion.py` converts a measurement in meters to miles, feet and inches (2)

- **Problem solving** – convert the steps below into a sequence of Python statements.

1. Create constants for the given data as follows:

```
METERS_IN_A_MILE = 1609.34

YARDS_IN_A_MILE = 1760

FEET_IN_A_YARD = 3

INCHES_IN_A_FOOT = 12
```

2. Read the measurement in meters and store the user input in a variable `meters`.

   Hint: call the function `input` to read in the user input as a string, and then call the function `float` to convert the string to a floating point number.

3. Convert the `meters` to miles, that is,

```
miles =  meters / METERS_IN_A_MILE
```

Program `Conversion.py` converts a measurement in meters to miles, feet and inches (3)

- **Problem solving** (continued)

4. Convert the `miles` to an integer using the function `int` and store the resulting integer in a variable `intMiles`. This gives you the **number of miles**.
5. Multiply the difference (`miles` – `intMiles`) by `YARDS_IN_A_MILE` and `FEET_IN_A_YARD`, and store the result in a variable `feet`.
6. Use a strategy similar to step 4 to convert the `feet` to an integer and store the integer in a variable `intFeet`. This gives you the **number of feet**.

   Use a strategy similar to step 5 to find the inches. You will need to replace the variables `miles` and `intMiles` with `feet` and `intFeet` respectively in the calculation, and then multiply the difference by `INCHES_IN_A_FOOT`. You can use the function `round` to return the value of inches rounded to a whole number.
7. Print the miles, feet and inches.

Program `Conversion.py` converts a measurement in meters to miles, feet and inches (4)

- Provide a comment at the beginning of the program to explain the purpose of the program, along with your name and the date.
- Save your program to the file `Conversion.py`
- Run your program.

Supplementary Questions for Private Study

- The laboratory worksheet contains supplementary questions in section 5 for private study.

- You are encouraged to complete the supplementary questions at home, or in the laboratory if you have time after completing questions 2 to 4.

Appendix A User input (PFE, Section 2.5 Input and Output)

- Use the function `input` to read keyboard input. For example,

```
firstname = input("Enter your first name: ")
```

- The function `input` displays the prompt (i.e. the string argument) in the *Shell* window and places the cursor on the same line, immediately following the string. See the output below.

```
Enter your first name:
```

- The program waits until the user types a name followed by the `Enter` key.

  The user input is stored as a string in the variable `firstname`. Recall that a string consists of a sequence of characters.

- To read an integer or floating-point value, use the function `input` followed by the function `int` or `float`.

  The function `int` converts the string into an integer, e.g.

```
noOfBooks = int( input("Enter number of books: ") )
```

  The function `float` converts the string into a floating-point value.

Appendix B Built-in function `int` (PFE, Section 2.5.2)

- `int(x)`

  The function `int` converts a number or string into an integer.

  Parameter: x A string or numerical value

  Returns: The new integer object

- First example to show the use of the function `int`:

```
userInput = "24"
# the function int converts the input string to an
# integer 24 and this is stored in the variable books
books = int(userInput)
```

- Second example to show the use of the function `int`:

```
price = 4.59
# the function int truncates the floating point value
# and returns the integer 4 in this example.
dollars = int(price)  # assign 4 to variable dollars
```