CSE 231 Fall 2009

Programming Project #1

Assignment Overview

This assignment involves coding and testing of a program based on the "Hello World" program from the first lab.

The basic design of the first programs that you construct in this class consists of a prompt for information, receiving information, processing that information then display the results.

This assignment is worth 10 points (1% of course grade), and must be completed before 11:59 PM on Monday, September 14th. Projects are typically due on Mondays.

Background

This programming project will use the `raw_input` and `print` functions along with some simple mathematics for conversion. The important part of the project is to learn the skills needed to access the class web site to download a project description, create a new program in Python and finally to hand it in.

We are going to estimate the population of the United States based on some information found at a website provided by the US census: http://www.census.gov/population/www/popclockus.html That website specifies the following information:

- A number that represents the current population. We will fix that number for the this project at the value: 307357870
- 3 rates:
    - every 7 seconds, a birth
    - every 13 seconds, a death
    - every 35 seconds, a new immigrant

Predicting into the future using only these numbers is not very accurate since the rates are likely to change, but it does give you a general idea for short predictions (only a few years into the future). Longer predictions (100's of years) are likely to be more inaccurate.

Program Specifications

Your program will prompt the user for an **integer** representing the number of **years** into the future we are predicting. Your program then prints out the predicted population size as an **integer**. You must use the above-indicated values: the starting point and the rates. The input value must represent years.

Use the following information about time conversion:

- 365 days in a year (not quite true, but let's use it anyway)
- 24 hours in a day
- 60 minutes in an hour
- 60 seconds in a minute

Deliverables

proj01.py -- your source code solution (remember to include your section, the date, project number and comments).

1. Please be sure to use the **specified file name**, i.e. “proj01.py”
2. Save a copy of your file in your CS account disk space (H drive on CS computers).
3. Electronically submit a copy of the file.

Assignment Notes:

To input the numbers it is necessary to use the `raw_input` function. The `raw_input` function takes a string, a sequence of characters between quotes, as a prompt to print to the user. It then waits until the user types a response, terminated by the user typing the Enter key. A string, again as a sequence of characters, is returned.

The returned string must be converted to a number. In this assignment we are strictly working with floating point numbers, a string is converted to an float using the `int` function. The `int` function takes as an argument a single string and returns integer number the string represents. A typical interaction would be something like:

```
numStr = raw_input(‘Please enter a number: ‘)
intVar = int(numStr)
```

`print` is a command that will print on the output window any combination of variables, values and strings. Each item to be printed must be separated from other items by a comma. All the items will be printed together, followed by a new line. For example:

```
billsInt = 345
print ‘The number ’,billsInt,‘ times two is ’, billsInt*2
```

This command has 4 items to print: a string (‘The number ’), the value in the variable billsInt (345), another string (‘ times two is ’) and the result of an expression (690). It will print:

```
The number 345 times two is 690
```

Look at the program numberInput.py in the proj01 directory as an example of using `raw_input, print` and `int` (also `float` for floating point numbers).

Once converted to numbers, the operations on these numbers are, respectively: + (sum), **-** (difference), ***** (product), / (division) and **%** (remainder). The last two deserve special comment.

In Python, if an integer is divided by another integer, the result is an integer. Thus the result of **6/4** is **1** (not 1.5). That is, the “/” operation results in the integer **quotient**. The result of **6%4** is the integer remainder of the division, thus **2** (6 divided by 4 is 1 with a remainder of 2). *Play around with the quotient and remainder operators in the Python shell window until you get comfortable with how it works.*

To clarify the problem specifications, we provide at the end of this document a snapshot of interaction with the already written program.

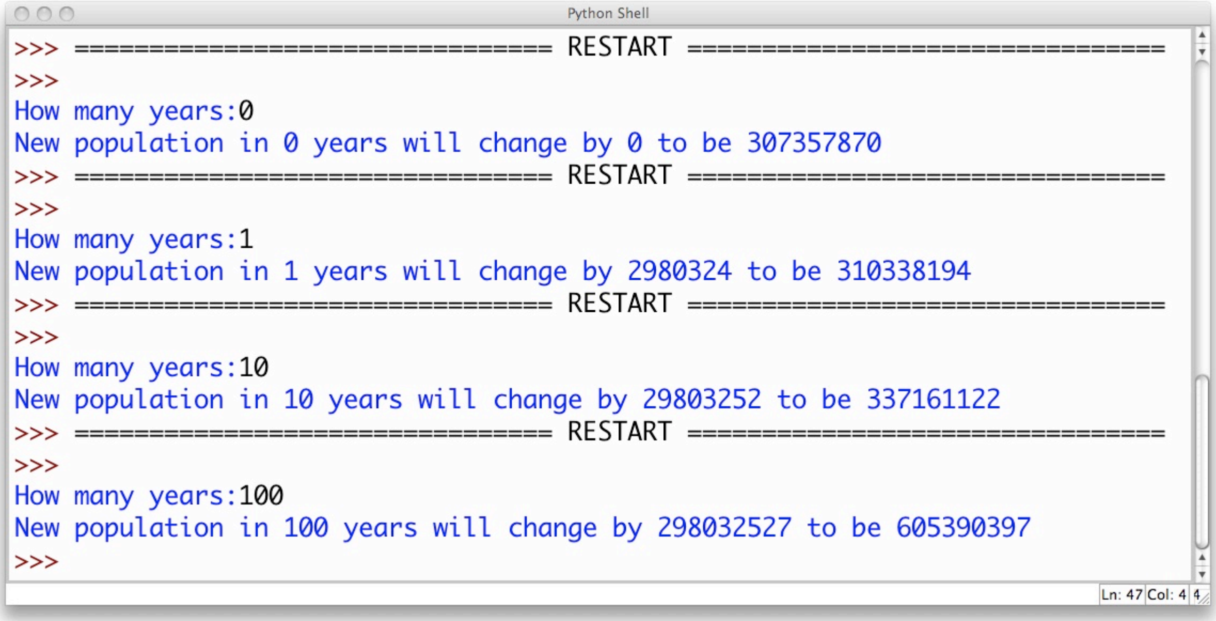
Getting Started

1. Using IDLE create a new program.
2. If you are in a CSE lab, select the H: drive as the location to store your file
3. Save the name of the project: proj01.py
4. Using the example from numberInput.py, write the code. Track down any errors (shouldn't be any at first).
5. Run the program
6. Use the web site to hand in the program (to make sure you can do it)
7. Edit the program
8. Now you enter a cycle of edit-run to incrementally develop your program.
9. Hand in your final version.

Questions for you to consider (not hand in)

1. What happens when you try to divide by zero when you run your program?
2. What happens when you enter a letter instead of a number at the prompt?

Sample Interaction



```
>>> ================================ RESTART ================================
>>> 
How many years:0
New population in 0 years will change by 0 to be 307357870
>>> ================================ RESTART ================================
>>> 
How many years:1
New population in 1 years will change by 2980324 to be 310338194
>>> ================================ RESTART ================================
>>> 
How many years:10
New population in 10 years will change by 29803252 to be 337161122
>>> ================================ RESTART ================================
>>> 
How many years:100
New population in 100 years will change by 298032527 to be 605390397
>>> 
```