

Programming Assignment 1 – Basic Probability, Computing and Statistics 2015

Fall 2015, Master of Logic, University of Amsterdam

Submission deadline: Monday, September 7th, 9 a.m.

Note: if the assignment is unclear to you or if you get stuck, do not hesitate to contact [Philip](#).

1 Exercise

This week we are going to build a first small interactive program. The program is a form of dialogue system. It should ask the user questions, store the answer and make inferences about the user based on his answers. Below, we show the system's desired output.

- *Hello, I am* some name of your choice. *What is your name? Please only give me one first name followed by your entire last name (e.g. Edward Snowden or Vincent van Gogh.)*
- *It's a pleasure to meet you. What is your birthday?* (specify input format here)
- *Do you smoke? (yes/no)*
- *Let me test your math skills. What is $\text{number1} * \text{number2}$?* Here number1 and number2 should be **random** integers between 1 and 1000 (both inclusive). This means that each time you run the program, this question will look slightly different. We have not discussed random numbers in class. Figure out yourself how to get them. A good starting point is [this page](#) in the Python documentation.
- Depending on the correctness of the answer, the program should say one of the following:
 - *Great, this is correct!*
 - *You got that one wrong. The correct answer is* correct answer. *Seems like you need to practice some more.*

- *It was nice talking to you. Let me summarise what I learnt about you. Your given name is the given name. Your surname is the surname. You are correct age years old. You are a smoker/non-smoker (depending on the information given by the user). You know how to do multiplication/You still need to practice multiplication.* (depending on whether the math exercise was solved correctly)

2 Grading

This section tells you what to look out for when programming. It also tells you how you should grade the assignment. In general, whenever you grade someone else's program, try to break it! This means that you should test limit cases in which the program might fail despite working correctly on more common inputs. In the present case this means that you should first test the program with true information about you but then also with information that might be true for other people. For example, identify yourself once as a smoker and once as a non-smoker. Also solve the math exercise correctly once and botch it the other time.

- 2 points All questions show up correctly and in the right order (give full credit when the program output is formulated slightly differently from the one here)
- 1 point The program correctly splits up the given name and surname. Check whether it can deal with complex surnames. A friend of mine is called Leria de la Rosa. The program should correctly identify such a surname.
- 2 points The program determines the correct age of the person at the time of execution. Play around with that. 1 point is to be deducted if the program cannot deal with people born b.c. (which we will denote by negative integers) or in the year 0.
- 1 point The program correctly identifies smokers and non-smokers. Crucially, it should not matter whether the answer is typed in upper or lower case.
- 1 point The math exercise changes its numbers at each execution.
- 1 point The numbers in the math exercise are indeed in the range [1,1000] and not in [0,999]. (This can be checked through a lot of repetition, but it might be better to do it analytically by inspecting the code).
- 1 point The program correctly computes the correct solution to the exercise.
- 1 point The program correctly identifies whether the solution to the math exercise given by the user is wrong or right.
- 2 points If the program is executable but crashes on any of the user inputs.