CS 171
Lab Assignment 1
Introduction to Python

This lab assignment use many elements provided in the main bibliographic reference for these lectures:
Programming in Python 3

A Complete Introduction to the Python Language,
2nd Edition,
Mark Summerfield

1 Obtaining and Installing Python3

If you have a modern Mac or Unix-like system, you may already have Python 3 installed.

You can check this by typing in a console:

```
$ python -V
```

If the version is 3.x you have already got Python 3 and don't have to install it yourself.

If Python was not found at all, it may have a name which includes a version number.

Try typing `python` and using tab to autocomplete the available options.

If still you can not a Python installation, you may find detailed installation instructions which are specific to your system at

`www.python.org/download`

2 Exercises

Exercise 1 *Write a program that determines the perimeter and the area of a circle whose radius is entered by the user. Please consider* `pi=3.1416`

An example of the execution of such program is:

```
$ Please enter the radius of the circle: 5
Perimeter = 31.416
Area = 78.54
```

□

Exercise 2 *Write a program that reads two numbers from the user, and checks whether the first is a multiple of the second.*

Two examples of the execution of such program are:

```
$ Please enter the first number: 336
$ Please enter the second number: 7
336 is a multiple of 7

$ Please enter the first number: 210
$ Please enter the second number: 9
210 is not a multiple of 9
```

It might be helpful to search for a description of the `%` *Python operator.*

□

Exercise 3 *Complete the following definition of the* `list_max` *function so that it takes a list of integers as argument and returns the greatest element in that list.*

```
def list_max(int_list):
    ...

print(list_max([1, 2, 8, 3, 10, 5]))
```

The output of the above program should naturally be

```
10
```

□

Exercise 4 *Improve your solution to Exercise 1 such that, if a non-numeric value is entered by the user, the program detects it and outputs an appropriate error message instead of crashing.*

The idea is to avoid program executions as the following, where the user inserted `d` *as input:*

```
$ Please enter the radius of the circle: d
Traceback (most recent call last):
  File "circle.py", line 2, in <module>
    r = int(input("Enter radius:"))
ValueError: invalid literal for int() with base 10: 'd'
```

And

```
$ Please enter the radius of the circle: d
You have not inserted a valid number!
```

□

Exercise 5 *Having reached this point, you are strongly encouraged to:*

1. *Study in detail the example programs that are described in Chapter 1, Section Examples, starting on page 39 of the reference book;*
2. *Provide solutions to the exercises in the same Chapter, Section Exercises, which start on page 47.*

□