Lab 6 – Functions and Modules

Deadline: Sunday, Oct. 19th 2025 11:59 pm
Summission

- `geometry.py`
- `main.py`
- `lab6_<your_name>_<SID>_q1.py`
- `lab6_<your_name>_<SID>_q2.py`

Part 1 – Lab Checkoff (5 pts)

Q1. (3 pts) Function practices

1. Write a function *argmax(l)*, which takes a single tuple argument, and return a tuple that consists of the maximum item in the type and its index. For example, *argmax*((1, 3, 2, −1, 4, 6, −7, −8)) will return (6, 5).

2. Write a function *mol()*, which can take arbitrary number (>=1) of arguments and return their product in float type.
For example, *mol(1.0)* will return 1.0, *mol(1.0, 1.5)* will return 1.5, and *mol(2.0, −1.0, 3.0)* will return −6.0.

3. A leap year is a year that can be divisible by 400, or that is not divisible by 100 but is divisible by 4. Write a function leap-year that does not take any input argument, and asks the user to input a start year and an end year, outputting a tuple that contains all the leap year between the start year and the end year, inclusive.
For example, one execution of print *leap-year()* can result in the following output.
Enter start year: 1998
Enter end year: 2009
(2000, 2004, 2008)

Q2. (2pts) Newton's method to calculate sqare root

Given a number **N**, write a function *sqrt_n()* to find the square root of that number using Newton's Method.
We talked about the Newton's method in the lab. Briefly, you start with almost any estimate x *(>0)*, and compute a better estimate y with the formula.

$$y = \frac{1}{2}(x + \frac{N}{x})$$

The process is repeated until the estimate stops changing, when we can a close-enough results (the difference between estimate < 1e-6)

```
# For Example
def sqrt_n(num):
    # function body

print(sqrt_n(6.0))
```

Part 2- Program Assignment (8 pts)

Q1. (3 pts) Function and Module

1) Create a module named *geometry.py*. This module should contain two functions: *rectangle_area()* and *circle_area()*, responsible for computing the area of a rectangle and a circle respectively.
2) Develop a separate program named main.py. In this program:
   a) Prompt the user to select a shape: either "rectangle" or "circle".
   b) Based on the chosen shape, ask the user to input the relevant dimensions. (For a rectangle, these would be length and width; for a circle, it would be the radius.)
   c) Using the functions from the geometry module, compute and display the area of the chosen shape.

For example:

Choose a shape (rectangle or circle): rectangle
Enter length: 10
Enter width: 5
The area of the rectangle is: 50.0

Q2. (2pts) Number conversion

Write a program that asks the user for an integer, and then displays its value in binary or Hexadecimal based on the user input.

Note:

- Each digit in a hexadecimal number ranges from 0-15. The value 10, 11, 12, 13, 14 and 15 are denoted with A,"B,"C", "D", "E" and "F", respectively.
- You can NOT use Python built in functions, such as *bin()* or *hex()*

```
# Test Case 1
Enter a number: 27
Convert to Binary or Hex: Hex
27 in Hex is: 1B

# Test Case 2
Enter a number: 9
Convert to Binary or Hex: Binary
7 in Hex is: 1001
```

Q3. (3pts) Numerical Integration of Gaussian function

We discussed how to calculate the function integration numerically in the lab. Write a function *gauss_integration()* to compute the integration of Gaussian function e^{-x^2} of a given range [a, b].

$$\int_a^b e^{-x^2} dx$$

Numerical integration is usually more accurate as the number of grid points increases. The accurate result of above expression in the range of [0, 1] is 0.74682413. Can you choose the number of grids so that the error < 0.0001?