KOÇ ÜNİVERSİTESİ

**Lab05**

COMP 125 Programming with Python

## Recording Disclaimer


The synchronous sessions are recorded (audiovisual recordings). The students are not required to keep their cameras on during class.

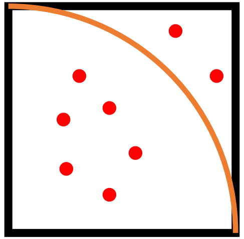
The audiovisual recordings, presentations, readings and any other works offered as the course materials aim to support remote and online learning. They are only for the personal use of the students. Further use of course materials other than the personal and educational purposes as defined in this disclaimer, such as making copies, reproductions, replications, submission and sharing on different platforms including the digital ones or commercial usages are strictly prohibited and illegal.

The persons violating the above-mentioned prohibitions can be subject to the administrative, civil, and criminal sanctions under the Law on Higher Education Nr. 2547, the By-Law on Disciplinary Matters of Higher Education Students, the Law on Intellectual Property Nr. 5846, the Criminal Law Nr. 5237, the Law on Obligations Nr. 6098, and any other relevant legislation.

The academic expressions, views, and discussions in the course materials including the audio-visual recordings fall within the scope of the freedom of science and art.

# Estimating the number Pi

- We will use the Monte Carlo integration to estimate the number $\pi$
- Assume that you have a square with edge length=1 and lower left corner at the origin.
- Using a random number generator, pick N random points (x,y) within the square ($0 \leqslant x \leqslant 1$ and $0 \leqslant y \leqslant 1$).
- Count the number of points inside the quarter circle (i.e. points with distance from origin less than 1).
- The estimate of $\pi$ is four times the ratio of this count over the total number of generated points



$$\pi \approx \frac{6}{8} \times 4 = 3$$

- Write a function named estimate_pi, which takes the number of random points N as an argument, and prints the estimate.
- Using this function estimate $\pi$ by using N=10, 100, 1000, 10000, … 10^8.

- Hint 1: To get random numbers in range [0,1) use random function: from random import random
- Hint 2: For the real $\pi$ use: from math import pi

# Estimating the number Pi

- Bonus: Modify your function such that it returns the estimate of $\pi$, rather than printing on the screen.
- Using this function calculate the error in estimated pi. How does the error change with increasing N?
- Modify your output to get the following, where error is abs(pi-pi_est)

```
N=        10 pi_est= 1.60000 error=  1.54159
N=       100 pi_est= 3.36000 error=  0.21841
N=      1000 pi_est= 3.15200 error=  0.01041
N=     10000 pi_est= 3.11320 error=  0.02839
N=    100000 pi_est= 3.13604 error=  0.00555
N=   1000000 pi_est= 3.14027 error=  0.00132
```

Hint: use function abs to calculate the absolute value

Hint: error = abs(pi - pi_est)

# Taylor series of $e^x$

- The Taylor series $e^x = 1 + x + x^2/2! + x^3/3! + x^4/4! + \ldots$
- We will use Taylor series to estimate $e^2$.
- Copy the factorial function discussed in class to your code.
- Write a function e_x(n), where n is the number of terms to be used in Taylor series specifically for calculating $e^2$. This function should use the factorial function for the calculation. It should print the number of terms used and the estimate of $e^2$ to the screen.
- Calculate the estimate of $e^2$ by using n=1, 2, ... , 10.

# Taylor series of $e^x$

- Bonus: Modify your function. It should take two arguments e_x(x,n) such that x is the point where we want to estimate $e^x$ and n is the number of terms to be used. It should return the estimate of $e^x$.
- Compare your estimate with the exact value of $e^x$

  (from math import exp; ex=exp(1)**2)

- Format your output such that you get:

```
N=  1 Estimate of e^x= 3.00000e+00 error= 4.38906e+00
N=  2 Estimate of e^x= 5.00000e+00 error= 2.38906e+00
N=  3 Estimate of e^x= 6.33333e+00 error= 1.05572e+00
N=  4 Estimate of e^x= 7.00000e+00 error= 3.89056e-01
N=  5 Estimate of e^x= 7.26667e+00 error= 1.22389e-01
N=  6 Estimate of e^x= 7.35556e+00 error= 3.35005e-02
N=  7 Estimate of e^x= 7.38095e+00 error= 8.10372e-03
N=  8 Estimate of e^x= 7.38730e+00 error= 1.75451e-03
N=  9 Estimate of e^x= 7.38871e+00 error= 3.43577e-04
N= 10 Estimate of e^x= 7.38899e+00 error= 6.13899e-05
N= 11 Estimate of e^x= 7.38905e+00 error= 1.00832e-05
N= 12 Estimate of e^x= 7.38905e+00 error= 1.53210e-06
N= 13 Estimate of e^x= 7.38906e+00 error= 2.16541e-07
```

Bonus to the bonus: After a certain N, the error does not decrease any more. Why?