

CS103 – Fall 2025 – Lab09

General Instructions

1. In today's lab, you are required to complete all required exercises during class to receive attendance credit.
2. You are welcome to:
 - Work together with classmates.
 - Search online for help or documentation.
 - Ask the TA for guidance if you are stuck.
3. To receive credit, you must finish and show your solutions to the TA before leaving lab.
4. There are also extra (optional) exercises at the end for those who want to challenge themselves. These are not required for attendance credit but are recommended for practice.

Exercise Instructions

- Make a folder **Lab09** inside your **cs103fa25** folder.
- Create a new python file inside your Lab09 folder (lab09.py).

Required Exercises

Exercises

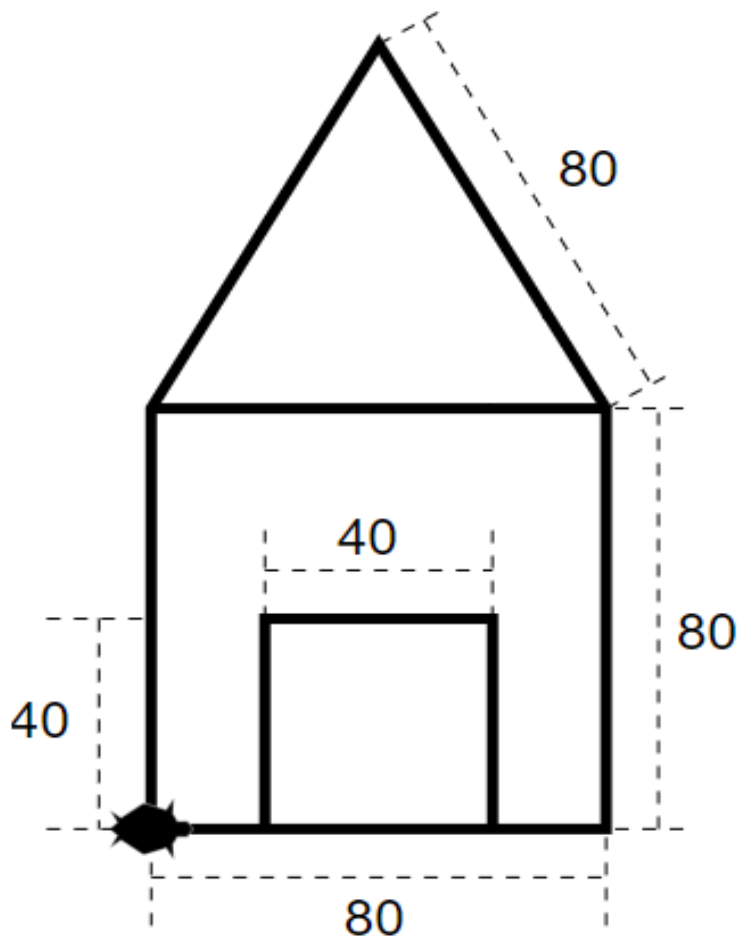
EXERCISE 1:

Write a function `tHouse` that takes an int `n1` and draws a basic house shape which has the lengths of `n1` pixels. The size of the gate will be $(n1/2) \times (n1/2)$ pixels. *Note: You will use turtle graphics to do this. It might help to create helper functions to accomplish this task.*

Sample Input:

```
n1 = 80
```

Sample Output



EXERCISE 2:

Write a **recursive** function called `fibonacci(n)` that returns the n th Fibonacci number.

Requirements

1. The function must use **recursion** (no loops or lists).
2. It must have proper **base cases** for when n is 0 or 1.

Example test

```
n = 6
print("Fibonacci number at position", n, "is", fibonacci(n))
>>>Fibonacci number at position 6 is 8
Hints: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
```

Extra (Optional) Challenges

PRACTICE PROBLEM 1:

Write a function **armstrongChecker** that takes an int "**a**" and returns a **Boolean Value** (True/False). The function checks whether a given number is an Armstrong number or not. An Armstrong number is a number such that the sum of its digits raised to the third power is equal to the number itself (Assume **a** is a three-digit number: 100-999).

Sample Input:

```
a = 371
```

Sample Output

```
True
```

```
Hints: 3**3+7**3+1**3=371.
```

PRACTICE PROBLEM 2:

Write a function “**cEmail**” that takes a string “**s1**” and returns another string. Assuming that **s1** is an email address in the “username@companyname.com” format, please write function to print the company name of a given email address. Both usernames and company names are composed of letters only.

Sample Input

```
s1= "unan@uab.edu"
```

Sample Output

```
"uab"
```

PRACTICE PROBLEM 3:

Write a function “**basicStats**” that takes a NumPy array “**npArray**” and returns a list. The list will contain three values: mean, standard deviation, and variance of the input array.

Sample Input:

```
npArray = [0,1,2,3,4,5]
```

Sample Output:

```
[2.5, 1.707825127659933, 2.9166666666666665]
```

hints:

- mean = 2.5
- std = 1.707825127659933
- var = 2.9166666666666665

To get attendance credit, finish Exercises 1 and 2.

If you finish early, try the optional challenges!