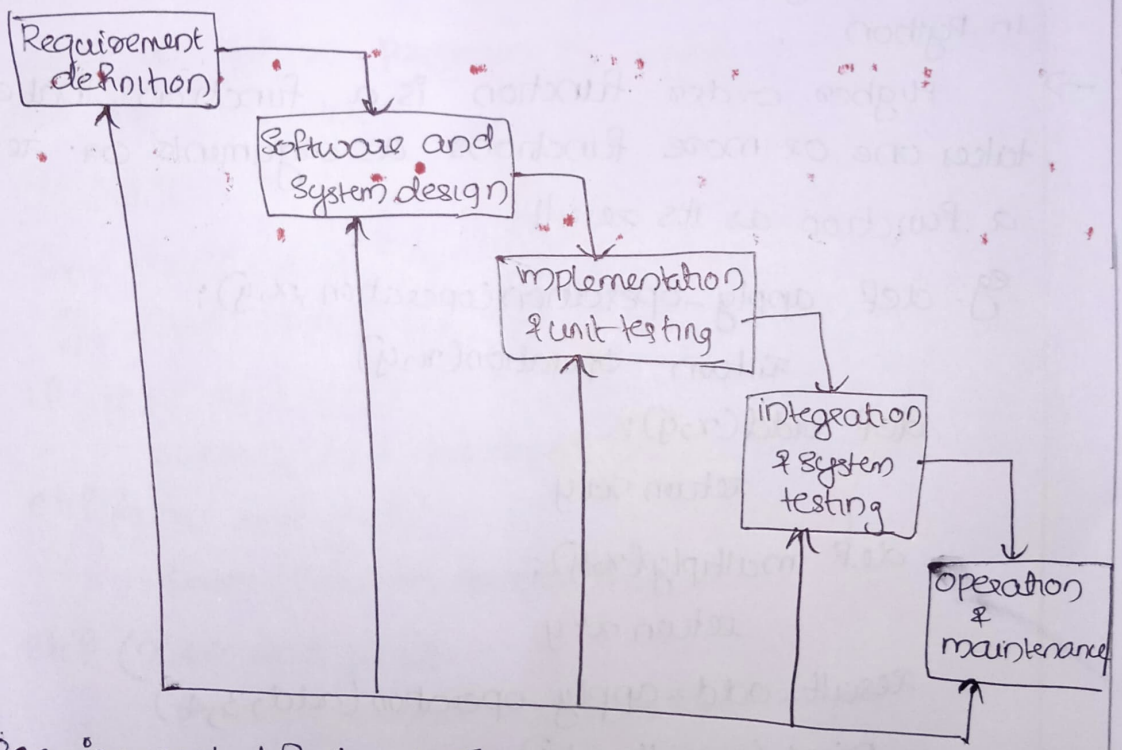
1. Which are the different stages in a software development process? Explain the waterfall model in detail.

→ The software development process typically involves stages like

- Requirements gathering: understanding and documenting project needs
- Planning: outlining tasks, timelines and resources
- Design: creating the system architecture and user interface
- Implementation - writing and testing the actual code.
- Testing - Assessing software to identify and fix issues.
- Deployment - Releasing the software for users.

waterfallmodel



(a) Requirement definition:- This system service, constraints and goals are established by consultation with system users.

(b) System and software design: Design process allocates the requirements to either hardware or software system.

(c) Implementation & unit testing :- During this stage the software design is realized as a set of programs or program units.

(d) Integration & system testing: The individual program or program units are integrated and tested as a complete system to ensure that software requirement have been met.

(e) Operational maintenance: The system is installed and put into the practical use maintenance involve correcting errors that were not discovered in the early steges of cycle.

2. Explain higher order functions and lambda functions in Python

→ Higher order function is a function that either takes one or more functions as arguments or return a function as its result

eg: def apply_operation(operation, x, y):
        return operation(x, y)

    def add(x, y):
        return x+y

    def multiply(x, y):
        return x*y

    result_add = apply_operation(add, 3, 4)
    print(result_add)
    result_multiply = apply_operation(multiply, 3, 4)
    Print(result_multiply)

1. Write a python program to input a time in seconds & print the time in HH:MM:SS format.

Solution:

```
timeInSeconds = int(input('Enter time in seconds'))

if timeInSeconds >0:
    hour = timeInSeconds // (60*60)
    minutes = (timeInSeconds % (60*60)) // 60
    seconds = (timeInSeconds % (60*60)) % 60

    hour = '{:02d}'.format(hour)
    minutes = '{:02d}'.format(minutes)
    seconds = '{:02d}'.format(seconds)

    Print('Time in HH:MM:SS format:', hour, ':', minutes, ':', seconds)
else:
    print('Invalid Input. Time in seconds must be positive!')
```

2. Write a Python program to check the validity of a password given by the user.

The password should satisfy the following criteria.

a) Contains atleast one letter between a & z
b) Contains atleast one number between 0 & 9
c) Contains atleast one letter between A & Z
d) Contains atleast one special character from $, #, @
e) minimum length of password: 8

Sol

```
def is_valid_password(password):
    if len(password) < 8:
        return False

    has_lowercase = False
    has_uppercase = False
    has_number = False
    has_special_char = False

    for char in password:
        if char.islower():
            has_lowercase = True
        elif char.isupper():
            has_uppercase = True
        elif char.isdigit():
            has_digit = True
        elif char in ['$', '#', '@']:
            has_special_char = True

    if has_lowercase and has_uppercase and has_number
     and has_special_char:
        return True
    else:
        return False

Password = input('Enter your password')

if is_valid_password(Password):
    print('Valid password')
else:
    print('Invalid Password')
```

9. Write a python program to do basic set operations.

9. Write a python program to do basic set operations.

Sol:

```
def basic_set_operations(set1, set2):
    Print('set 1:', set1)
    Print('set 2:', set2)

    Print("In Union of set1 &set2 is: ", set1.union(set2))
    Print("Intersection of set1 &set2 : ", set1.intersection(set2))
    Print("Difference of set1 &set2 : ", set1.difference(set2))
    Print("Symmetric Difference of set1 and set2: ", set1.symmetric_
                                                    difference(set2))
set1_input = input("Enter Elements of set1 seperated by spaces: ")
set2_input = input("Enter elements of set2 seperated by spaces: ")

set1 = set1_input.split()
set2 = set2_input.split()
set1 = {int(ele) for ele in set1}
set2 = {int(ele) for ele in set2}
basic_set_operations(set1, set2)
```