P1:

1. Create a design specification for data structures
   1. What is ADT?
   2. How to implement ADT?
2. explaining the valid operations that can be carried out on the structures.
   1. What Common Operations Can Be Carried Out on the ADT
   2. Get an example about an implement ADT: Linked List. All of the set of operations And the main to show the result.

P2:

1. Determine the operations of a memory stack
   1. What is a stack? Definition, operation, exception.
   2. How to implement stack? How many ways to do this?
      1. More details about the way you choose.
2. how it is used to implement function calls in a computer.
   1. What is a function call in a computer? How does it work?
   2. How can stack work in Function Calls? ((with illustration).

P3:

1. What is an imperative ADT or imperative definition in ADT?
2. What is Software Stack? (List three or five software stacks)? How they implement ADT?

M1:

1. Introduce about queue:
   1. What is a queue?
   2. Which operations on a queue
   3. Queue exceptions | Pros and Cons of Queue
   4. Applications of Queue
2. How to implement a queue?
3. Show an illustration of using a queue | Code

M2:

1. What is sorting algorithms? Classify them.
2. Briefly describe two sorting algorithm you compare, show the implementation and comparison .

M3:

1. What is Encapsulation and Information Hiding?
2. the advantages of encapsulation and information hiding when using an ADT.

D1:

1. What is the shortest path problem?
2. Analyze and illustrate two shortest path algorithms. (algorithm and illustration)

D2:

Discuss about the state: “imperative ADTs are a basis for object orientation”. Explain your opinion.