

Algorithms & Data Structures

Lab 9 (Queue Implementation in Python)

```
# The Queue_class simulates a Queue
class Queue_class:
    # The __init__ method initializes the
    # Queue data attribute
    def __init__(self,size):
        self.queue = []
        self.size = size

    # The enqueue method add an item to the Queue
    def enqueue(self, item):
        if len(self.queue) == self.size:
            print("Queue is full!!")
        else:
            self.queue.insert(0, item)
            print('Queue size is: ', len(self.queue))

    # The dequeue method delete an item from the
    queue
    def dequeue(self):
        if self.queue == []:
            print("Queue is empty!")
            data = -1
        else:
            data = self.queue.pop()
        return data

    # The display method will state items
    # of the queue
    def display(self):
        if self.queue == []:
            print("Queue is empty!")
        else:
            print("Queue data:")
            print("Front ==> ",self.queue," <==
Rear")
```

```

# Let the user set the size of the queue
s = int(input('Enter the size of the Queue'))
exit = False

# Create an object from the Queue class.
# Queue (with a capital Q) is the class name
# queue (with a small q) is the object
# created from the "Queue" class
queue = Queue_class(s)
while exit is False:
    # Choose which operation you want
    # to apply to the queue.
    print("1) Add")
    print("2) Delete")
    print("3) Display")
    print("Choose an operation: ")
    operation = input()

def addOp():
    number = input("Insert a number:")
    if number.isdigit():
        # we use the keyword "global" to force
        # the function to access
        # the main queue object that we created.
        global queue
        # call the function "enqueue" that we
        # defined in the main "Queue" class
        queue.enqueue(number)
    else:
        print("Invalid input...")

def delOp():
    global queue
    n = queue.dequeue()
    if n != -1:
        print(f"Deleted data: {n}")

```

```

def displayOp():
    global queue
    queue.display()

def exitOp():
    global exit
    exit = True
    print("Exiting...")

# Create a dictionary
# Dictionaries are data structures special
# to python they save values as a pair
# all other data structures store values one
# at a time.
# dictionaries save values as (Key:Value)
switch = {
    '1': addOp,
    '2': delOp,
    '3': displayOp
}
# "get" is a special function related to
dictionaries.
# When get() is called, Python checks if the
specified
# key exists in the dict. If it does, then get()
returns
# the value of that key. If the key does not
exist,
# then get() returns the value specified in the
# second argument
switch.get(operation, exitOp())

```