In [66]:

```
## Task 1
class Queue:

    def __init__(self):
        self.queue = list()
    def insert(self, item):
        self.queue.append(item)
    def pop(self):
        if Queue.isEmpty(self):
            print("Queue is empty !")
            return None
        self.queue.remove(self.queue[0])
        return self.queue

    def print_queue(self):
        print(self.queue)

    def isEmpty(self):
        if len(self.queue)==0: return True
        else: return False
que1 = Queue()
que2 = Queue()
que1.insert(2)
que1.insert(2)
que2.insert(2)
que2.insert(2)
que2.pop()
que2.pop()
# print the message queue is empty
que2.pop()
que1.print_queue()
que2.print_queue()
```

```
Queue is empty !
[2, 2]
[]
```

In [65]:

```
# Task 2
import pickle
class AdvancedQueue(Queue):
    num_of_queue =0
    instances =list()
    def __init__(self,name ,size):
        super(AdvancedQueue).__init__()
        self.name = name
        self.size = size
        self.queue = list()
        AdvancedQueue.num_of_queue+=1
        self.__class__.instances.append(self)

    def insert(self, item):
        if len(self.queue) >= self.size:
            raise  Exception("Queue is full , Out Of Range size")
        return super().insert(item)

    def pop(self):
        return super().pop()

    def print_queue(self):
        return super().print_queue()


    @classmethod
    def GetQueueUsingName(cls,name):
        for que in cls.instances:
            if que.name == name:
                return que
    @classmethod
    def save(cls):
        pickle.dump(cls.instances, open("queueFile.pickle","wb"))

    @classmethod
    def load(self):
        queues = pickle.load(open("queueFile.pickle","rb"))
        return queues



# create a queue instances
q1 =AdvancedQueue("firstQueue",3)
q2 =AdvancedQueue("secondQueue",3)
q1.insert(3)
q1.insert(2)
q1.insert(4)



q2.insert(1)
q2.insert(4)

# print the number of instance of class
print("the number of instance that created from Advanced queue class : ",q1.num_of_queue)

q1.print_queue()
# get queue instance using name of queue
AdvancedQueue.GetQueueUsingName("firstQueue").print_queue()
# save the queue instance in file
AdvancedQueue.save();
# load the queue instance from file
data =AdvancedQueue.load();
print(data);
# should throw an exception because
# q1.insert(1)
```

```
the number of instance that created from Advanced queue class :  2
[3, 2, 4]
[3, 2, 4]
[<__main__.AdvancedQueue object at 0x000002119E5AF280>, <__main__.AdvancedQueue object at
0x000002119E6EA310>]
```

In [ ]:

```

```