Scala Queue

Queue implements a data structure that allows inserting and retrieving elements in a first-in-first-out (FIFO) manner.

In scala, Queue is implemented as a pair of lists. One is used to insert the elements and second to contain deleted elements. Elements are added to the first list and removed from the second list.

Scala Queue Example

```
import scala.collection.immutable._

object MainObject{

   def main(args:Array[String]){

      var queue = Queue(1,5,6,2,3,9,5,2,5)

      var queue2:Queue[Int] = Queue(1,5,6,2,3,9,5,2,5)

      println(queue)

      println(queue2)

   }

}
```

Output:

```
Queue(1, 5, 6, 2, 3, 9, 5, 2, 5)

Queue(1, 5, 6, 2, 3, 9, 5, 2, 5)
```

Scala Queue Example 2

```
import scala.collection.immutable._

object MainObject{

   def main(args:Array[String]){

      var queue = Queue(1,5,6,2,3,9,5,2,5)

      print("Queue Elements: ")
```

```
        queue.foreach((element:Int)=>print(element+" "))
        var firstElement = queue.front
        print("\nFirst element in the queue: "+ firstElement)
        var enqueueQueue = queue.enqueue(100)
        print("\nElement added in the queue: ")
        enqueueQueue.foreach((element:Int)=>print(element+" "))
        var dequeueQueue = queue.dequeue
        print("\nElement deleted from this queue: "+ dequeueQueue)
    }
}
```

Output:

```
Queue Elements: 1 5 6 2 3 9 5 2 5
First element in the queue: 1
Element added in the queue: 1 5 6 2 3 9 5 2 5 100
Element deleted from this queue: (1,Queue(5, 6, 2, 3, 9, 5, 2, 5))
```