Java 2 – Week 9 Assignment – Stacks and Queues

Instructions

Create a Java application that demonstrates the use of Stack and Queue data structures. The application should be able to:

- Perform basic operations (push, pop, peek, etc.) on a stack.
- Perform basic operations (enqueue, dequeue, peek, etc.) on a queue.
- Display the contents of the stack and queue at any point.

TO DO:

1. Create a new Java project and setup the main class.
2. **Implement a Stack**:
    - Create a class MyStack using generics.
    - Implement methods: push(), pop(), peek(), isEmpty().
    - Write a method to display the elements of the stack.
3. **Implement a Queue**:
    - Create a class MyQueue using generics.
    - Implement methods: enqueue(), dequeue(), peek(), isEmpty().
    - Write a method to display the elements of the queue.
4. **Create a Main Class**:
    - Use the main class to demonstrate the functionality of both the stack and qu eue.
    - Interact with the user to perform operations and display the results.

Start Code:

```
import java.util.LinkedList;

class MyStack<T> {
  private LinkedList<T> stack = new LinkedList<>();

  public void push(T element) {
    stack.push(element);
  }

  public T pop() {
    return stack.pop();
  }

  public T peek() {
    return stack.peek();
  }

  public boolean isEmpty() {
    return stack.isEmpty();
  }

  public void display() {
    System.out.println("Stack: " + stack);
  }
}

class MyQueue<T> {
  private LinkedList<T> queue = new LinkedList<>();

  public void enqueue(T element) {
    queue.addLast(element);
  }

  public T dequeue() {
    return queue.pollFirst();
  }

  public T peek() {
    return queue.peekFirst();
  }

  public boolean isEmpty() {
    return queue.isEmpty();
  }

  public void display() {
    System.out.println("Queue: " + queue);
  }
}

public class Main {
  public static void main(String[] args) {
    MyStack<Integer> stack = new MyStack<>();
```

<add code here>

```
  }
}
```

Add:

- Extend the stack to implement a min() function that returns the minimu m element in constant time.

When completed submit on Blackboard for grading. Make sure to include the code and output.