```java
package com.mycompany.app;

import java.time.Duration;
import java.util.Arrays;
import java.util.LinkedList;
import java.util.Properties;
import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.apache.kafka.clients.consumer.ConsumerRecord;
import org.apache.kafka.clients.consumer.ConsumerRecords;
import org.apache.kafka.clients.consumer.KafkaConsumer;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.apache.kafka.common.serialization.StringSerializer;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import java.util.Queue;

public class Consumer {

    /**
 * Class: 44-517 Big Data
 * Author: Gabriel ("Makerspace Manager") Solomon Holland
 * Description: This is a kafka multi channel consumer project
 * Due: November 11, 2022
 * I pledge that I have completed the programming assignment independently.
   I have not copied the code from a student or any source.
   I have not given my code to any other student.
   I have not given my code to any other student and will not share this code
   with anyone under any circumstances.
*/
    public static void main(String[] args)
    {
        Logger logger = LoggerFactory.getLogger(Consumer.class.getName());
        String bootstrapServers = "localhost:9092";

        // create Consumer properties
        Properties properties = new Properties();
        properties.setProperty(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
bootstrapServers);
        properties.setProperty(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class.getName());
        properties.setProperty(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class.getName());
        properties.setProperty(ConsumerConfig.GROUP_ID_CONFIG, "dreloemisleadme");

        KafkaConsumer<String, String> consumer = new KafkaConsumer<>(properties);


        //subscription (wow even open source projects are making you subscribe,
disgusting)

        //uncomment which ones you're using
        //consumer.subscribe(Arrays.asList("Rack1Temps"));
        consumer.subscribe(Arrays.asList("Rack2Temps"));
        //consumer.subscribe(Arrays.asList("Smoker"));
        int count = 0;

        Queue<Double> temps = new LinkedList<Double>(); //This is a QUEUE, gib me
bonus points
        Queue<String> times = new LinkedList<String>();
```

```
55
56          Double doubValue = 0.0;
57          String[] recordArr;
58          Boolean stall;
59          //Boolean giveAttention;
60
61          while(true) { //while true *vomit emoji*
62              ConsumerRecords<String, String> records =
consumer.poll(Duration.ofMillis(100));
63              for (ConsumerRecord<String, String> record: records)
64              {
65
66                  //split the line we're on
67                  recordArr = record.value().split("\t");
68                  doubValue = Double.parseDouble(recordArr[1]);
69
70                  //System.out.println(record.topic() + ", " + record.key() + ", " +
record.value());
71                  temps.add(doubValue);
72                  times.add(recordArr[0]);
73
74                  //pull off the head and assign to the var we already made. No sense
in making more vars.
75                  doubValue = temps.peek();
76
77                  /*
78                  //this is for the smoker attention code
79                  //reset bool value for multiple runs
80                  giveAttention = false;
81
82                  for(Double i:temps)
83                  {
84                      //if temps drop 15 degrees over 2.5 minutes (5 datapoints)
85                      if(i>=doubValue+15 || i<=doubValue-15)
86                      {
87                          giveAttention = true;
88                          break;
89                      }
90                  }
91
92                  if(giveAttention)
93                  {
94                      System.out.println("Give attention at: " + times.peek());
95                  }
96                  if(temps.size()>5)
97                  {
98                      //remove oldest (first value) to not save entire dataset
99                      //System.out.println(temps);
100                     temps.remove();
101                     times.remove();
102                 }
103                 */
104
105
106
107                 //this is for stall code
108                 //reset bool value for multiple runs
109                 stall = true;
110
```

```
111                 //parse through our array, max size should be 20. 20 size = 10
    minutes
112                 for(Double i:temps)
113                 {
114                     //if any value is NOT within 1, it sets our check to false
115                     if(i>=doubValue+1 || i<=doubValue-1)
116                     {
117                         stall = false;
118                         break;
119                     }
120                 }
121                 if(stall)
122                 {
123                     System.out.println("Stall Starts at:  " + times.peek());
124                 }
125 
126 
127                 if(temps.size()>20)
128                 {
129                     //remove oldest (first) value because we don't want to save the
    entire data set
130                     temps.remove();
131                     times.remove();
132                 }
133 
134 
135             }
136         }
137 
138     }
139 
140 }
141 
```