Answers

1. Closing the file is necessary so we will not miss the updates (adds, changes, deletes) we have made to it.

2.

```
import java.io.*;
import java.io.FileReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
import java.io.BufferedReader;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

public class StringArrayChallenge {
 public static void main(String[] args) {
  /**
   Read from the file provided, AMC.csv.

   Print out the date and value of the highest high and lowest low.
  */

  try {
   String file = "AMC.csv";
   FileReader fileRead = new FileReader(file);
   BufferedReader input = new BufferedReader(fileRead);

   String[] rowInformation; //use rowInformation to store the whole line of the file with split format
   String line; //use line to get the value in the file

   line = Files.readAllLines(Paths.get(file)).get(1); //read the second line which is also the first line with numbers
   rowInformation = line.split(","); //split the whole line with comma and store them into the array

   double maximum = Double.parseDouble(rowInformation[2]);   //record the highest high and initialize with the first high
value
   double minimum = Double.parseDouble(rowInformation[3]);   //same but with first low value
   String dateMaxi = rowInformation[0]; //initialize the dateMaxi with the first date
   String dateMini = rowInformation[0]; //same

   //check the value one by one until the end of file
   while((line = input.readLine()) != null){
    int i = 0;
    line = input.readLine();
    rowInformation = line.split(",");

    //record the current high, low and date
    double currentHigh = Double.parseDouble(rowInformation[2]);
```

```
        double currentLow = Double.parseDouble(rowInformation[3]);
        String currentDate = rowInformation[0];

        // if current value greater than highest, store it and date to the related value
        // if lower than lowest, store it
        if (currentHigh > maximum){
          maximum = currentHigh;
          dateMaxi = currentDate;
        } else if(currentLow < minimum){
          minimum = currentLow;
          dateMini = currentDate;
        }
      }

      //print the related message
      System.out.printf("The highest high in the file is: %f on %s\n", maximum,dateMaxi);
      System.out.printf("The lowest low in the file is: %f on %s\n", minimum,dateMini);

      //close them
      fileRead.close();
      input.close();

    }
    catch (IOException e){
      System.out.println("Could not read from file");
    }
  }
}
```