1.a) No, Our program Lab2Part5c do not do any error checking. We have not implemented any error handling for this.

1.b) Please see file Lab3Part1b.java for this question.

1.b.i ) I used if statement to compare each and every value with every other value. In this way, if any of the conditions fails, then the program will throw an error that the values are not in order and exit there itself.

if(exactStartTime >exactEndTime || exactStartTime >exactBreakStartTime ||

exactStartTime >exactBreakEndTime || exactBreakStartTime >exactEndTime

||exactBreakStartTime >exactEndTime || exactBreakStartTime >exactEndTime)

{

System.out.println("Values are not in Order...");

System.out.println("Exiting Program...");

return;

}

1.b.ii) We can also do the same task by making an array of the values and sorting that array. Then we can compare our four values with the corresponding values in the array. If they don’t match then it means we have an error;

int[] arr=new

int[]{exactStartTime,exactBreakStartTime,exactBreakEndTime,exactEndTime};

Arrays.sort(arr);

if(arr[0]!=exactStartTime || arr[1]!=exactBreakStartTime || arr[2]!=exactBreakEndTime || arr[3]!=exactEndTime)

{

System.out.println("Values are not in Order...");

System.out.println("Exiting Program...");

return;

}

This method of testing will involve sorting and a new array. So it will be costlier in terms of memory as well as time taken.

1.b.iii) Another approach could be simply taking the absolute value of the time instead of taking 0 as the hours. Because sometimes the minus sign can be added by mistake. So we can simply keep the hours and minutes values same and take their absolute values to make them positive.

Time=Math.abs(Time);

1.c) Please see file Lab3Part1c.java for this question.

1.c.i) If our program reads the time inputs as strings, it will never give us InputMismatchException as a string can contain all type of values. It may be integer values, double values or even String values.

1.c.ii) Other types of data such as Names are also written as a combination of first name and last name; we can also separate it by using space as a delimiter. We can use scanner delimiter ‘ ’ in this case to read the values.

Other data like DOB is also written in a particular format. It might be DD/MM/YYY or any specified format. We can use scanner delimiter ‘/’ in this case to read the values.

1.d) Please see file Lab3Part1d.java for this question.

1.d.i) We chose this loop structure.

while ((sCurrentLine = br.readLine()) != null)

{

process(i,sCurrentLine);

i++;

}

We chose this loop because we will have to traverse through whole input file and traverse each line. This loop is traversing through all the input file line by line till there are no files to read.

1.d.ii) There are various benefits of reading over a file than to read from the user input file.

Some benefits of reading from file:

* We can read the data that was previously stored. I mean there might be old data that needs to be processed.
* In case of logs, the logs can be easily analyzed for different types of errors that occurred.
* 3.It is very cumbersome and lengthy to type all the data into the console. We can prepare the input files in parallel and then process them at once.

The drawback of reading data from a file :

* data is not validated in files. So that might produce various runtime errors.

1.e) Please see file Lab3Part1e.java for this question.

1.e.i)Writing this method for rounding gives us great benefits.

* We can simply write one method and then it will be used for all the calculations of that type. Here, I have written one method, and it is being called all the times for performing the rounding task.
* We have to write less code that is reusability. We are using the same method. So it is making our code modular and easy to use.

1.e.ii) There are various other sections of the code that can we made modular. I have already made these sections modular as we have to read the file and then process each read line. We also have to validate the entered military time. I already have used methods for all these tasks.