**Programming Assignment:** You work as an independent software programmer. One of your clients who owns a payroll company wants you to write a program to identify bad records in their employee database.

These employee records are kept in a text file called “employee.txt”.

NOTE: Your program should handle the case where this file is not found. You should display an appropriate message and shut the program down if this happens.

Your job is to write a program that parses this file and identifies the good and bad records. You must then write the good records to a file called “good.txt” and write the bad records to a file called “bad.txt”.

The file *employee.txt* may be found in the following directory:

*Google Drive:/AP Computer Science/Programming Projects/exceptionHandlingLab/employee.txt*

Each line of this file contains a record for one employee. The typical line looks like this:

***020-12-4567:John Smith:46000***

Each field is separated by a colon (so the colon is a delimiter). Each line is supposed to have three fields - a Social Security Number, the employee’s name, and their salary.

The file *employee.txt* contains many lines such as these. Most of the records are fine. But a few are corrupt. Some have a bad SSN, while others may be missing one or more fields. The name field must be a non-empty string while the salary must be a positive integer.

**Program Requirements**

As you read (parse) each line, your program should generate (throw) exceptions when a bad employee record is detected. These bad records need to be copied into the bad.txt file. If a record is fine, it needs to be copied to the file good.txt.

Your program should also throw an exception if the input file cannot be located. For this case, it should print a warning to the console and shut the program down.

After your program finishes, it should close all three files. You need not throw exceptions for these actions if they fail.

One piece of code is being made available to you to help you:

Method public boolean validSSN(String ssn) takes a String argument and returns true if the given string is a valid SSN and false otherwise. The code for validSSN() can be found in the same directory as noted above.

**Split() Method May Be Useful**

You may also find it useful to use the String method split() to split each line into separate fields (SSN, name, and salary). Here is an example of how the split method can be used to split a line into separate fields with a colon serving as a delimiter.

String assetClasses = **"Gold:Stocks:Fixed Income:Commodity:Interest Rates";**

String[] splits = asseltClasses.split(":");

System.out.println("splits.size: " + splits.length);

for(String asset: splits){

System.out.println(asset);

}

**OutPut**

splits.size: 5

Gold

Stocks

Fixed Income

Commodity

Interest Rates