Exercise 5 (10 points) – can be done individually or in pair

- The first lines of all source files must be comment containing names & IDs of all members. Also create file readme.txt containing names & IDs of all members.
- Put all files (source, input, output) in folder Ex5_xxx where xxx = your full ID. That is, your source files must be in package Ex5_xxx and input/output files (if there is any) must be read from/write to this folder. From now on, you’ll get point deduction for wrong package & folder structure.
- The group representative zips Ex5_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted.
- The exercise is graded only once, and after graded, members can’t be added.

==

Modify your code from Exercise 4 to handle exceptions. All reused code from Exercise 4 must be submitted with Exercise 5 (I won't go back to get it from your previous submission).

1. Handle exceptions when opening file i.e. when file is not found. Keep asking user for a valid file name.

2. Handle exceptions when user inputs non-numeric value for threshold year. Keep asking user for a valid input.

3. Put the code that reads & splits each line of data in try-catch. If a runtime exception occurs, print exception message and line that causes the exception. Then, skip that line and read the next one. Take not of which exception occurs in which situation, as this will be asked in Midterm exam.

4. Use code from Exercise 4 to select, sort, and print Company objects as before. Due to the exceptions in (3), some of them will be excluded from the output.

```
PepsiCo,              l965,   260,    8,   88          l instead of one
Santander,            1857,    63,   1O,   87          O instead of zero
Walmart,             -1962,   400,   12,  611          invalid year  (negative)
Citigroup,            2998,    90,   15,  120          invalid year  (exceed current year)
Tencent Holdings,     1998,  -415,   27,   82          invalid input (negative)
TD Bank Group,        1855,   114;   11,   57          semi-colon instead of comma
BP                    1909,    90,   26,  248          missing comma
Cisco Systems,        1990,   189,   19                missing column
Equinor,              1972,    90,   29,  142, 53      exceeding column
American Express,     1850,  11.4,    7,   68          double instead of int
Microsoft,          -19.75,  2310    6S,  208          multiple exceptions
```

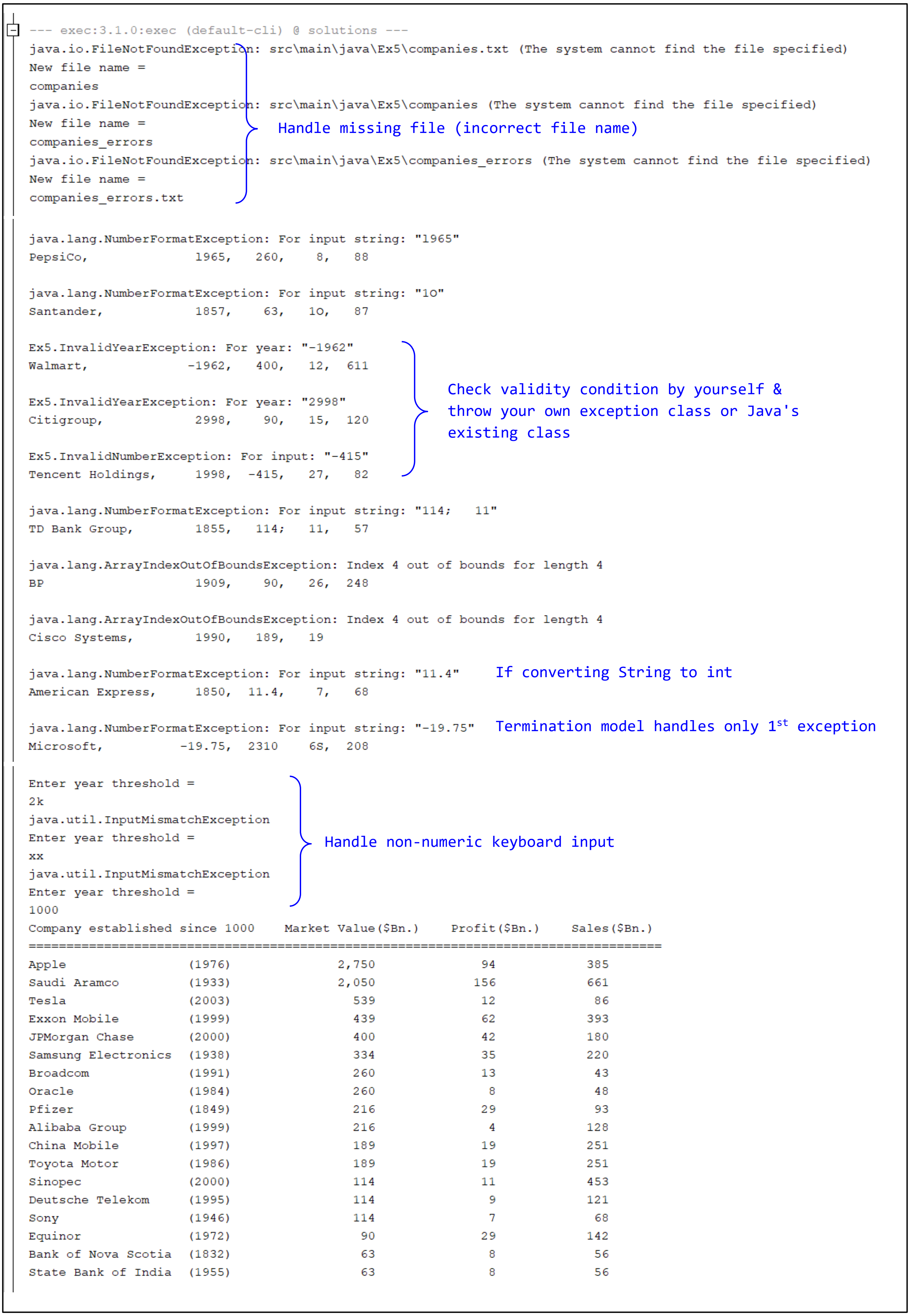
Note 1 : Some conditions such as invalid year/number will not cause runtime exception. You have to check these conditions & throw exceptions by yourself.

```
if (value < 0)  throw new MyException("Invalid " + value);     // your own class
if (value < 0)  throw new IllegalArgumentException(value);     // Java class
```

Note 2 : Some conditions such as exceeding column will not cause runtime exception and have to effect on the calculation or result. You can just ignore it, i.e. let the program read only columns 0-4.

Note 3 : Some conditions such as input being double instead of int may or may not cause runtime exception, depending on how you handle input String e.g.

- Convert String to int directly
- Convert String to double, then cast double to int



```
--- exec:3.1.0:exec (default-cli) @ solutions ---
java.io.FileNotFoundException: src\main\java\Ex5\companies.txt (The system cannot find the file specified)
New file name =
companies
java.io.FileNotFoundException: src\main\java\Ex5\companies (The system cannot find the file specified)
New file name =
companies_errors
java.io.FileNotFoundException: src\main\java\Ex5\companies_errors (The system cannot find the file specified)
New file name =
companies_errors.txt
```

Handle missing file (incorrect file name)

```
java.lang.NumberFormatException: For input string: "1965"
PepsiCo,              1965,   260,    8,   88

java.lang.NumberFormatException: For input string: "10"
Santander,            1857,    63,   10,   87

Ex5.InvalidYearException: For year: "-1962"
Walmart,             -1962,   400,   12,  611

Ex5.InvalidYearException: For year: "2998"
Citigroup,            2998,    90,   15,  120

Ex5.InvalidNumberException: For input: "-415"
Tencent Holdings,     1998,  -415,   27,   82
```

Check validity condition by yourself & throw your own exception class or Java's existing class

```
java.lang.NumberFormatException: For input string: "114;   11"
TD Bank Group,        1855,   114;   11,   57

java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
BP                    1909,    90,   26,  248

java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
Cisco Systems,        1990,   189,   19

java.lang.NumberFormatException: For input string: "11.4"
American Express,     1850,  11.4,    7,   68
```

If converting String to int

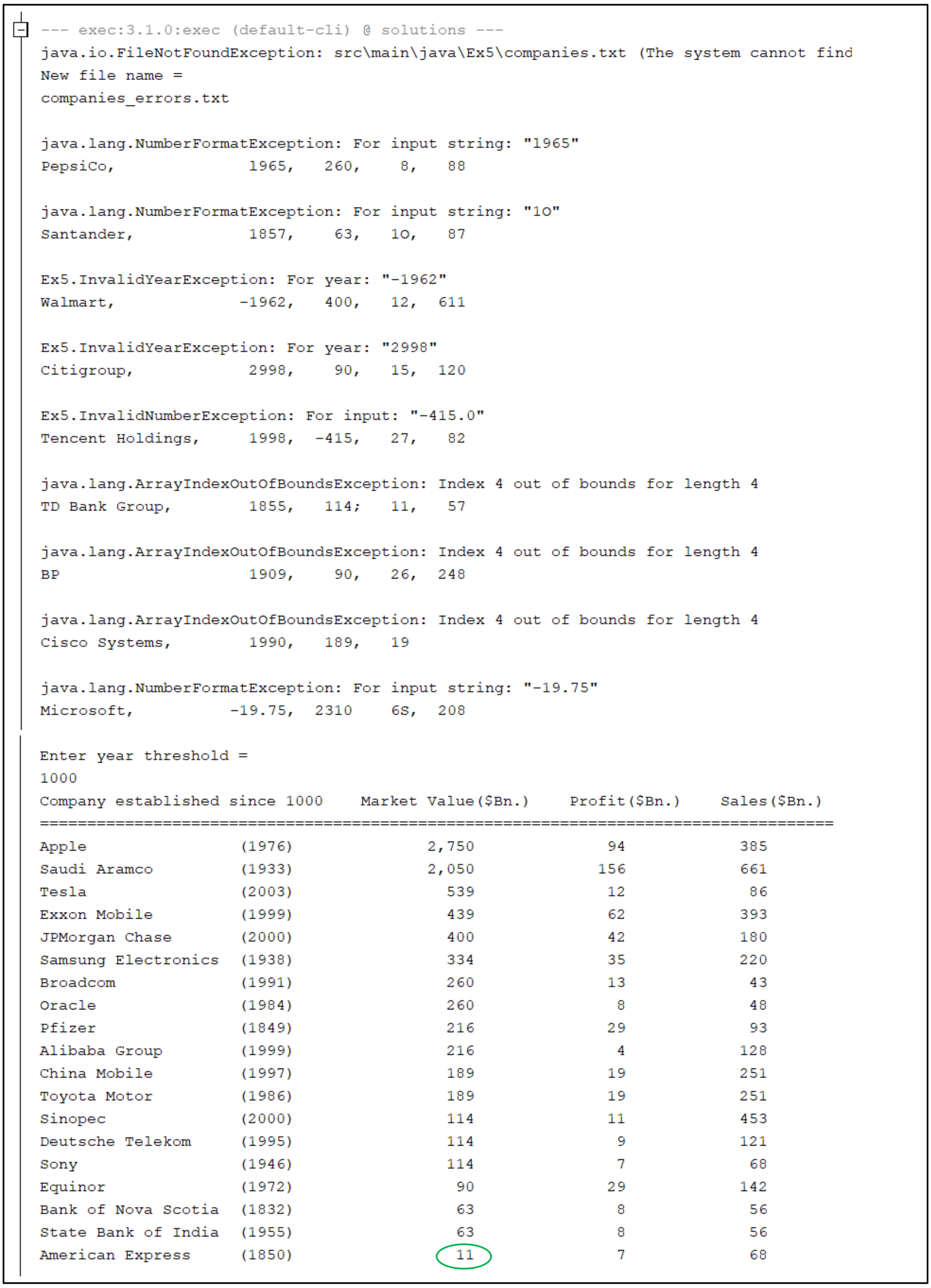
```
java.lang.NumberFormatException: For input string: "-19.75"
Microsoft,          -19.75,  2310    6S,  208
```

Termination model handles only 1st exception

```
Enter year threshold =
2k
java.util.InputMismatchException
Enter year threshold =
xx
java.util.InputMismatchException
Enter year threshold =
1000
```

Handle non-numeric keyboard input

```
Company established since 1000    Market Value($Bn.)    Profit($Bn.)    Sales($Bn.)
====================================================================================
Apple                (1976)              2,750              94            385
Saudi Aramco         (1933)              2,050             156            661
Tesla                (2003)                539              12             86
Exxon Mobile         (1999)                439              62            393
JPMorgan Chase       (2000)                400              42            180
Samsung Electronics  (1938)                334              35            220
Broadcom             (1991)                260              13             43
Oracle               (1984)                260               8             48
Pfizer               (1849)                216              29             93
Alibaba Group        (1999)                216               4            128
China Mobile         (1997)                189              19            251
Toyota Motor         (1986)                189              19            251
Sinopec              (2000)                114              11            453
Deutsche Telekom     (1995)                114               9            121
Sony                 (1946)                114               7             68
Equinor              (1972)                 90              29            142
Bank of Nova Scotia  (1832)                 63               8             56
State Bank of India  (1955)                 63               8             56
```



```
--- exec:3.1.0:exec (default-cli) @ solutions ---
java.io.FileNotFoundException: src\main\java\Ex5\companies.txt (The system cannot find
New file name =
companies_errors.txt

java.lang.NumberFormatException: For input string: "l965"
PepsiCo,              l965,   260,    8,   88

java.lang.NumberFormatException: For input string: "1O"
Santander,            1857,    63,   1O,   87

Ex5.InvalidYearException: For year: "-1962"
Walmart,             -1962,   400,   12,  611

Ex5.InvalidYearException: For year: "2998"
Citigroup,            2998,    90,   15,  120

Ex5.InvalidNumberException: For input: "-415.0"
Tencent Holdings,     1998,  -415,   27,   82

java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
TD Bank Group,        1855,   114;   11,   57

java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
BP                    1909,    90,   26,  248

java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
Cisco Systems,        1990,   189,   19

java.lang.NumberFormatException: For input string: "-19.75"
Microsoft,          -19.75,  2310    6S,  208

Enter year threshold =
1000
Company established since 1000    Market Value($Bn.)    Profit($Bn.)    Sales($Bn.)
====================================================================================
Apple                (1976)              2,750              94            385
Saudi Aramco         (1933)              2,050             156            661
Tesla                (2003)                539              12             86
Exxon Mobile         (1999)                439              62            393
JPMorgan Chase       (2000)                400              42            180
Samsung Electronics  (1938)                334              35            220
Broadcom             (1991)                260              13             43
Oracle               (1984)                260               8             48
Pfizer               (1849)                216              29             93
Alibaba Group        (1999)                216               4            128
China Mobile         (1997)                189              19            251
Toyota Motor         (1986)                189              19            251
Sinopec              (2000)                114              11            453
Deutsche Telekom     (1995)                114               9            121
Sony                 (1946)                114               7             68
Equinor              (1972)                 90              29            142
Bank of Nova Scotia  (1832)                 63               8             56
State Bank of India  (1955)                 63               8             56
American Express     (1850)                 11               7             68
```

11.4 is parsed as double, then casted to int