Practical-Exam-04: Sorting and Searching Algorithms

Due 18 Oct by 12:00 **Points** 100 **Available** 18 Oct at 10:15 - 18 Oct at 12:00 about 2 hours

This assignment was locked 18 Oct at 12:00.

Introduction

This session ends 12 pm

Make sure to complete your submission in time.

Submit to SVN & Websubmission (for every problem solved)

Note that on the top of each file you will see an indication whether any change is necessary or not.

Submission

- Create the repository on the SVN server
- You must checkout only the folder practical-exam-04 from your server
- No other folders from your SVN will be allowed during the practical exam.
- Check if your repository is being track by WebSubmission before start solving the exam.

```
svn mkdir -m "first assessment commit" --parents https://version-control.adelaide.edu.au/svn/<YOUR-ID>/2019/s2/fcs/week-10/practical-exam-04

```

Assessment

- **This is a practical exam - your work must be entirely your own.**
- Marks for this practical exam contribute 2 marks.
- Marks will be awarded later by your workshop supervisor (40%) and websubmission marker (60%).
- **Due date: the end of this section (12 pm).**
- **Do Not Forget** To Submit on **WebSubmission**
- **Late penalties:** Only work submitted during your enrolled practical session from a Linux system in the practical lab will be marked.

Regarding functional marks, please consider:

```
(1) only codes that can compile will be marked;
(2) only codes that are in the suggested directory will be marked;
(3) only codes submitted to SVN and WebSubmission before the deadline will be marked;
```

```
(4) only codes containing your signature on the top of the file will be marked by tutors;
(5) you will have your markers decreased in 3 points if *.class file present in your folders;
```

Note

- To acquire full marks **(1)** all your functionalities must be working perfectly, **(2)** your code has to be well and proportionally commented, and **(3)** your code must follow correct indentation (4 spaces for every new indentation level) and **(4)** you have to use all the content from latest lectures.
- We argue that you are not just asked to solve a problem but use the more sophisticated way to solve it. For instance, you can solve a problem using ten variables, but it will always be better to solve the same problem with an array.

Practical Exam 04

Download: project.zip

```
How to unzip "project.zip"

1. Move project.zip from Download folder to ...week-10/practical-exam-04/
2. In the terminal,
    2.1 Navigate to practical-exam-04
    for instance: cd ...assessment-04/practical-exam-04/

    2.2 In the folder practical-exam-04, create a folder "project"
    mkdir project

    2.3 Unzip project.zip into;
    unzip project.zip

    2.4 Remove project.zip
    rm project.zip

    2.5. Move all files into project/
    mv * project/

3. Submit your progress to SVN;
```

Hungry Donalds Fast Food

Dear students,

In this practical exam, you are a developer that works for the company **FCS-SA** Systems. Your company's main client is the fast food chain Hungry Donalds. This morning, your IT manager emailed you a new requirement for the system and a few issues reported by your company's client.

Please, **(1)** see attached manager email, **(2)** fix the issues reported and **(3)** write a reply email to your manager with a clear explanation of the problems and how you fixed;

Client email (see also in src/user_email.txt)

COMPOSE

Inbox (1)
Starred
Important
Sent Mail
Drafts (4)
Categories

Google Calendar
Today (Thu, 10 May)
12:00 [bill] $470 rent
Tomorrow (Fri, 11 May)
14:10 FCS Weekly Mee
Sat, 12 May
18:00 [party] Keijie Li di
Sun, 13 May
Mother's Day
Today Add Options

---------- Forwarded message ----------
From: **Beyonce** <rbeyonce@hungry.com>
Date: 10 May 2018 at 08:00
Subject: Re: Branch 551 - Critical error
To: Rafael Felix <rafael.felixalves@fcssa.edu.au>

Dear IT department,

I am the manager of Hungry Donalds (branch 551), MTV Land, SA - Australia. My code is a1145469.

This morning, after I logged in and load my staff members in the system I notice a series of abnormal behaviors.

The abnormal **problem (1)** consists of the functionality *"3. Sort staff by (name)"*.
This functionality should sort by staff member based on their names. However, every time I choose this option I need to restart the system due a strange error:

Error message: Exception in thread "main" java.lang.UnsupportedOperationException: Not implemented yet!

The abnormal **problem (2)** happens when I try to use any of the searching options.

> For instance, today I tried a search "6. Search staff by (age)". However, this functionality only works if I first do the functionality "4. Sort staff by (age)". If I "4. Sort staff by (age)", then "6. Search staff by (age)" the system works fine.

age: 14 then message: ":: JustinBieber 14 20 5000.0 1"

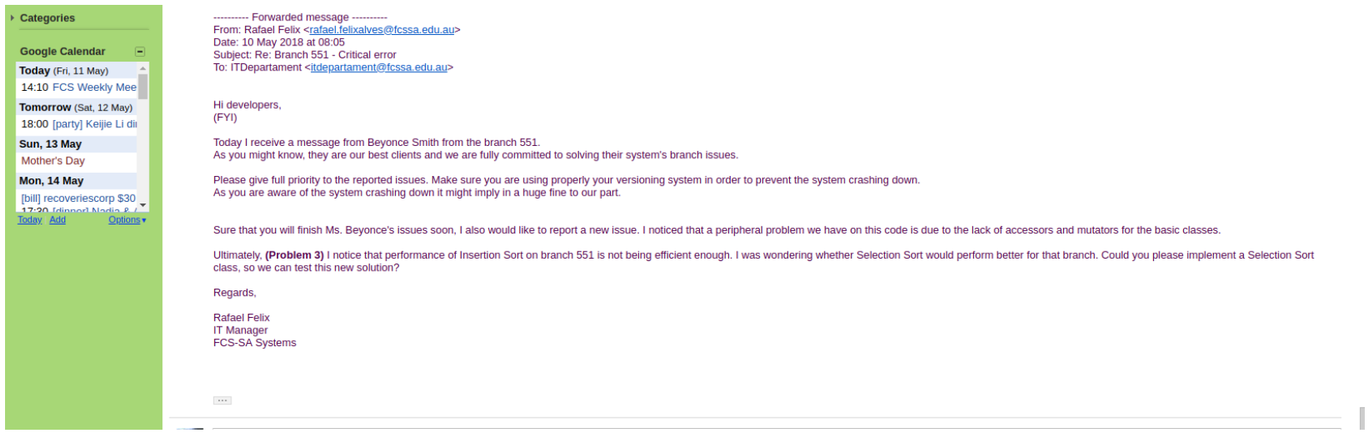
> Otherwise, it returns a strage message

age: 14, strange message: ":: Staff does not exist, try again!".

Considering that we will open the restaurant in 2h, I hope your team can solve these issues quicly.

Kind regards,
Beyonce Smith
Hungry Donalds Manager

IT Manager Email (see also in src/it_management.txt)



Reply to your 'email' to:

src/your_explanation_email.txt

*Do **NOT** open your email or send anything. Simply write your response in the text file as below*

```
1 Dear manager,
2
```

```
The problem (1), was due to <your explanation>, I fixed doing <your explanation>;
The problem (2), was due to <your explanation>, I fixed doing <your explanation>;
The problem (3), was due to <your explanation>, I fixed doing <your explanation>;

Kind regards, <your name> <your id>
```

You will be marked on this response. Even if you have not completed all the problems, you should write the 'email' and commit it to SVN as you make progress.

Instructions to run the code:

1. Download project.zip
2. Extract project.zip into your local repository;
3. Your final SVN repository must be organized as " . . . week-10/practical-exam-04/project
4. Navigate to your project folder;
5. Run:

    javac Main.java
    java Main

Instructions for use the software:

1. Function (1): load staff members, you MUST run it as the first action when starting the system.;
2. Function (2): display staff members. After loading staff members, you can display their info on the screen;

Other functions: check Menu::Help;

Problem 01: Sorting Staff by Name

Please, you are asked to follow the client requirement on the **problem (1),** all the information is reported on the email. In additional you are also asked to implement accessors and mutators for basic classes;

The built-in String method **.compareTo()** may be helpful.
https://www.tutorialspoint.com/java/java_string_compareto.htm (http://www.tutorialspoint.com/java/java_string_compareto.htm)

If there are two strings, *s1* and *s2*. Then *s1.compareTo(s2)* will give:

```
if s1 == s2 : 0
if s1 > s2: an int more than 0
if s1 < s2: an int less than 0
```

```
'''
```

```
## Signatures:

On this practical exam, you are NOT asked to design any signature other than accessors and mutators;
In addition, every file in the project.zip has a hint whether you need to perform changes or not.

## Requirements:
1. Set accessors and mutator for all properties;
2. Resolve the problem (1) reported by our client;
3. Do not crash the system;
```

```
## Repository

save this project and it's files in project/
add to your svn repository

Note: keep a track of your version. By the end of the session time you MUST submit a copy that can be
compiled.


## Important
1. Your classes MUST NOT contain a public static main function.
2. You MUST NOT perform changes on the files:
    a. BinarySearch.java
    b. LinearSearch.java
    c. Search.java
    d. Main.java
    e. Sort.java
    f. SortInterface.java
3. If you want to perform Unit Test, please use the class Test.java
'''
```

Problem 02: Fixing the behavior when Searching for staff members

Please, you are asked to follow the client requirement on the **problem (2),** all the information is reported on the email.

```
'''

## Signatures:

On this practical exam, you are not asked to design any signature other than accessors and mutators;
In addition, every file in the project.zip has a hint whether you need to perform changes or not.

## Requirements:
1. Resolve the problem (2) reported by our client;
2. Do not crash the system;

## Repository
```

```
save this project and it's files in project/
add to your svn repository

Note: keep a track of your version. By the end of the session time you MUST submit a copy that can be
compiled.


## Important
1. Your classes MUST NOT contain a public static main function.
2. You MUST NOT perform changes on the files:
    a. BinarySearch.java
    b. LinearSearch.java
    c. Search.java
    d. Main.java
    e. Sort.java
    f. SortInterface.java
3. It's likely that the bug in this code is hidden in the file Company.java; There are several ways o
f solving this issue, feel free to use your creativite.
4. If you want to perform Unit Test, please use the class Test.java
'''
```

Problem 03: Implement new features (Selection Sort)

To acquire marks: you need **sortIntByIndex**, the other methods are optional.

Please, you are asked to follow the manager requirement on the **problem (3),** all the information is reported on the email.

```
'''

## Signatures:

On this practical exam, you are not asked to design any signature other than accessors and mutators;
In addition, every file in the project.zip has a hint whether you need to perform changes or not.

## Requirements:
1. Resolve the problem (3) required by your manager;
2. Do not crash the system;

## Repository

save this project and it's files in project/
add to your svn repository

Note: keep a track of your version. By the end of the session time you MUST submit a copy that can be
compiled.
```

Important

1. Your classes MUST NOT contain a public static main function.
2. You MUST NOT perform change on the files:
    a. BinarySearch.java
    b. LinearSearch.java
    c. Search.java
    d. Main.java
    e. Sort.java
    f. SortInterface.java
3. It's likely that the bug in this code is hidden in the file **Selection.java** and **Company.java**;
4. If you want to perform Unit Test, please use the class Test.java

'''

Basic Marking Scheme (2)

Criteria	Ratings	Pts
Compilation In order to achieve full marks - your code must compile and run;		20.0 pts
Basic Functionality Your code (1) perform all the functions correctly, (2) use latest concepts learned in class, (3) has a clear, creative and sophisticated way of solving the problem.		30.0 pts
Functionality Extension Your code (1) perform all the functions correctly, (2) use latest concepts learned in class, (3) has a clear, creative and sophisticated way of solving the problem, and (4) you propose novel ways to solve the problems - or extra functionalities.		10.0 pts
Code Formatting Your code (1) has the right proportion of comments and place line and block comments correctly, (2) follow correct indentation every new indentation level, (3) has good variable naming, (4) has clear organization between tabs and is easy to read.		40.0 pts
		Total points: 100.0