

# CSC 241 Assignment 4

## Abstract Data Types and Programming Methodology

### Due: Thursday, March 31

## 1 Introduction

In this semester, we are going to develop a Java program called “*GradeManager*”, which manages students’ grades. This program provides functions such as adding/deleting/editing grades of certain students. All the grades are stored in designated file(s) and updated as grading data change. It also supports to manage students, *e.g.*, adding a new student and his/her grading data or deleting a grading record of a particular student. This program will be built up through several assignments, in each of which you will be asked to apply what you will learn in lectures. At the end of semester, you will have a Java program which utilizes various OOP techniques and diverse data structures.

The assignments will be handed in an order for completing a final program. So, you **MUST** follow instructions and achieve requirements when you work on an assignment. Java code for each assignment should be errorless and submitted in the Blackboard course shell. Since a next assignment usually asks you to add more functions or edit what have been made in the prior assignment, you should keep the previous Java code(s). If the prior program is submitted with errors or runs unsuccessfully, it must be corrected before it goes to the next assignment.

## 2 Goal for This Assignment

The aim of the assignment 3 was to load grading data of students in Json file into objects of Course/Section/Student class. In the assignment 4, you will develop methods for searching information from the data. For this, you will implement two searching algorithms – linear search and binary search that you learned in class. To develop such methods, you may use methods such as `equals( )` or `compareTo( )` which were introduced in lecture 5 and 6. You may also refer to the example codes used in class. Once the two searching methods are implemented, they will be tested and competed to show which algorithm is overwhelming. Through this benchmarking, you will prove how to analyze algorithms.

## 3 Instructions

### A. Template file

Each assignment should be built in a package. The names of package and class for this assignment are below.

**Package:** Assignment4

**Class:** GradeManager

```
package Assignment4;

...

public class GradeManager {
```

...

Java files and data files are zipped under the name of this package. You first unzip the **Assignment4.zip** which is uploaded in the Blackboard course shell, and check all necessary files. DO NOT change the package name or class name if you want to use the template files.

However, you may want to use the files you previously made, since this assignment is extended from the prior assignment. In this case, please duplicate them and start working on them. Do not forget that change the name of the package.

### B. Data File / Properties File

In this assignment, the Json file has more data. It includes 20 students. Therefore, you should make sure that there are data for 20 students in Json file you are loading while you are working on this assignment.

From the assignment 3, the file path is obtained from `config.properties` file. It is going to be used again in this assignment and the following assignments. You do not have to change it if it worked well in the assignment 3. However, if you want to run your program in the different system, you should update it for the new system.

### C. Developing Environment

Your program should be **implemented in Java only**. The program in another language will not be graded.

### D. Submission

You will submit your Java package. Zip the package **Assignment4** again and upload it in the Blackboard course shell. DO NOT copy and paste your code into text files such as rtf, doc, or txt. You **MUST submit java files**, not text files! The assignment will give you **two weeks** plus one more week due to the Spring Recess, so it is **due on Thursday, March 31**. All submission **by 11:59 PM** on that day will be accepted without any penalty. On the due date, Blackboard may be suffering of too much network traffics and be unstable. There is no excuse about the issue, therefore you are strongly recommended to submit earlier than the due date.

## 4 Requirements

### A. Complete all the requirements in the assignment 3

Regardless of new requirements for this assignment, your program should be able to do what were requested in the assignment 3. If you already completed all requirements in the assignment 3, you would concentrate on the new requirements. Otherwise, you must complete it first.

### B. Implement two searching algorithms + benchmark

The purpose of this assignment is to implement benchmark two searching algorithms. For this, you should implement the searching methods first, and then conduct benchmarking.

The benchmark progresses by counting the number of repetitions for each searching algorithm. The benchmark enabled searching will start when you enter a new menu “find,” shown in **Figure 1**. Once the menu is selected, a key must be input. Key could be any of information for the student, i.e., id, name, q1, q2, and so on. But in this assignment, you use only “name”. **Figure 1** shows a find request where the key is “name,” and the value is “Rachel.”

```
Enter a course code: CS241
Name: Abstract Data Types and Programming Methodology | CRN: 14607
| Code: ccs241 | Capacity: 24 | Time: 13:50
Select menu [find | edit | quit]? find
Enter what you want: name, Rachel
Name: Rachel | ID: 555333111 | Q1: 8 | Q2: 7 | Q3: 5 | Midterm: 90
| Final: 91
[Benchmark] 17 comparisons by Linear Search, 4 comparisons by
Binary Search
Select menu [find | edit | quit]? edit
Enter a student: John
Name: John | ID: 123456789 | Q1: 10 | Q2: 8 | Q3: 8 | Midterm:87 |
Final: 91
Enter a coursework you want to edit (q1,q2,q3,mid,final): q1
Enter a new score: 9
Name: John | ID: 123456789 | Q1: 9 | Q2: 8 | Q3: 8 | Midterm:87 |
Final: 91
Select menu [find | edit | quit]? quit
```

**Figure 1.** Your program has three menu – find, edit, quit.

For benchmark, a utility class is provided, named `Utility.Benchmark`. You have to download `Utility.zip` from the blackboard, unzip it and place it in the same directory of your working directory. Those who are using IntelliJ will put it in the outside of the directory “src”.

Once it is in the proper location, you are able to import it in a file you want to use.

```
import Utility.Benchmark;
```

The class is public and has static variables and methods, which are used for benchmarking the searching methods. They are listed below.

```
public static int counterLS;
public static int counterBS;
public static void setCounterLS(int number);
public static void resetCounterLS();
public static void increaseCounterLS();
public static void decreaseCounterLS();
public static int getCounterLS();
public static void setCounterBS(int number);
public static void resetCounterBS();
```

```
public static void increaseCounterBS();  
public static void decreaseCounterBS();  
public static int getCounterBS();
```

In the template package for assignment 4, you may see an example which uses the benchmark methods. Take a look at it and try it for your program.

### C. Adapt a searching algorithm

When a student's grade needs to be edited, searching algorithm should run first two times – to search the student and to find the course work which will be edited. From the benchmark for searching algorithms, you have witnessed that binary search usually had fewer comparisons than linear search, which means binary search predominant on average. Thus, you will apply binary search when you edit a grade for a student. Once you implement the binary searching method, you call it for the edit menu.

## 5 Grading

### A. Grading criteria

The lab is assigned **30** points, which is 10% of the final grade. It will be graded by evaluating the requirement. Any missing and unsatisfiable criteria will take off points. The tentative and brief criteria are below.

- Compilation: **5** points
- Execution: **5** points
- Proper output: **20** points

### B. Late penalty

Late submission will take off **10% per day** after due date. **Thus, submission after 10 days will not be accepted in any circumstances.**

## 6 Academic Integrity

Any dishonest behaviors will not be tolerated in this class. Any form of plagiarism and cheating will be dealt with according to the guidelines on the Academic Integrity Policy online at <http://www.oswego.edu/integrity>. For more information about university policies, see the following online catalog at:

[http://catalog.oswego.edu/content.php?catoid=2&navoid=47#stat\\_inte\\_inte](http://catalog.oswego.edu/content.php?catoid=2&navoid=47#stat_inte_inte)

**Student who is against the honor code will not have any credits in this project.**