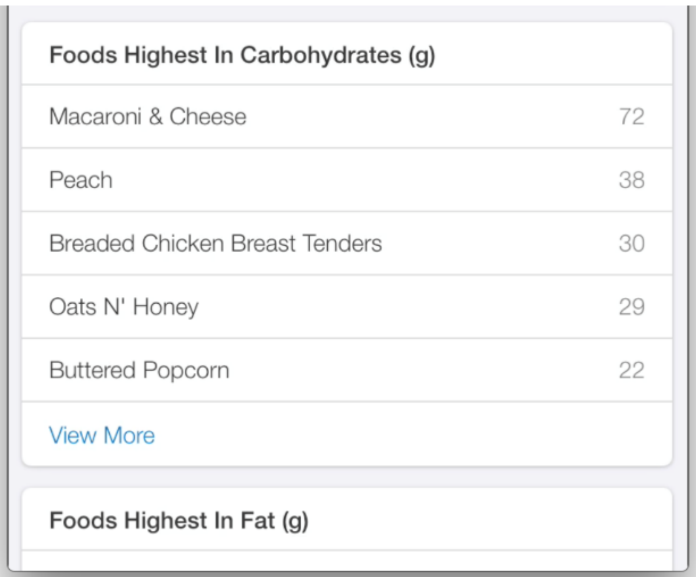
Project 1 – MSCI 240, Fall 2016[1]

Assigned: Friday, September 16, 2016
Part 1 Due: Friday, September 30, 12:20pm.
Part 2 Due: Friday, October 7, 12:20pm.
Value: This is a project. It will be worth 4-5% of your final mark. Marked out of 100pts.

There are a lot of recent apps on smartphones (and the web) for tracking health information, such as what food you eat, or what fitness activities you're doing. After entering this information over a long period of time, it's sometimes useful to be able to see a record of the highest-calorie foods or the fastest 5 runs that you took. Imagine you are building an app that needs to keep track of the top 5, 10, or 15 foods in terms of carbohydrates (see figure). This kind of list is always in sorted order, with the food with the highest calorie count at the top of the list, and the food with the lowest calorie count at the bottom. In some apps this may be 5 foods, and in other apps it may be 10, or any number chosen by the app designer. The figure at the right shows an app (MyFitnessPal) that shows the top 5 foods highest in carbohydrates, fat, etc.



Foods Highest In Carbohydrates (g)	
Macaroni & Cheese	72
Peach	38
Breaded Chicken Breast Tenders	30
Oats N' Honey	29
Buttered Popcorn	22
View More	

Foods Highest In Fat (g)

Source: http://myfitnesspal.com

In this project, you are to design and implement a Java class that can be used by other developers to implement a list of foods highest in **calorie** count. The requirements of this class are:

1. The class must have some mechanism to retain its state between invocations of the program. The usual way to do this is to have the class store its state to a file on the hard drive.
2. Default behavior of the class is to store up to 10 foods and calorie counts. The user of the class should be able to create a top food list object that stores from 1 to n top scores. In other words, the default constructor should make a list that has 10 slots for food names and calorie counts, and you should also create a constructor that allows the user of the class to specify the number of slots.
3. The top food list starts empty (i.e., no food has been reported). The class provides a means for food and calorie count pairs to be added to the list. If the new pair is a high-calorie food and belongs in the top n, it is placed in the correct place in the list. If more than n food and calorie count pairs are added to the table, only the top n are retained.
4. The class should provide a means for the user of the class to iterate through the foods and their calorie counts in sorted order.
5. The design of the class should follow good design principles and should properly encapsulate its implementation (make use of private and public appropriately).

[1] This project is a modified version of a previous project by Mark Smucker, used with permission

A Multi-Part Assignment

This assignment will consist of 2 parts that are turned in at different times. The parts are listed below.

Part 1 –Write the interface and test code and a report on both - 50 pts

For Part 1, you are to write the Java top food list class' public methods, but you are not required to implement the methods. For example, in Lab 1, you saw public methods without implementation as follows (not all parts of the class are shown):

```java
public class Fraction {
    public Fraction(String fraction) {
        // TODO
        // To be written by you, delete the following statement.
        throw new UnsupportedOperationException();
    }

    public static Fraction divide(Fraction lhs, Fraction rhs) {
        // TODO
        // To be written by you, delete the following statement.
        throw new UnsupportedOperationException();
    }
}
```

Please do the same for your class.

In addition, please create a **test plan** and corresponding **test code** to allow you to carry out the test plan. A test plan is the plan that you intend to follow to determine if your class works correctly. The test code is used to help you determine if your class works properly. Specifically, you need to develop a set of tests that will allow you to demonstrate that your class works correctly. The test plan will list each test, how it should work, and what requirement(s) of the class it is testing.

Write a report regarding part 1 of this project. Your report should be typed and consist of the following parts:

1. A cover page that includes the following information:
    a. Your name.
    b. The course info: "MSCI 240 Fall 2016, Instructor: Mark Hancock"
    c. The date on which you are submitting your assignment.
    d. Which assignment this is.
2. A report consisting of the following sections:
    a. Introduction: A brief summary of the whole report.
    b. The design of the public interface of your class. This should be the Java code including good documentation of the expected input and output of each constructor and public method.
    c. The test plan and its corresponding test code.
    d. Acknowledgments. You must acknowledge all help received. You must cite and quote all sources of material that you did not create yourself.
    e. Help Given. Please explain any help that you gave to other students.

Part 2 – Final Code and Report on Part 2 – 50pts

Implement, test, and debug your class until you believe it is working correctly. Visit the TAs and instructor for help.

Write a report regarding part 2 of this project. Your report should be typed and consist of the following parts:

3. A cover page that includes the following information:
    a. Your name.
    b. The course info: "MSCI 240 Fall 2016, Instructor: Mark Hancock"
    c. The date on which you are submitting your assignment.
    d. Which assignment this is.
4. A report consisting of the following sections:
    a. Introduction: A brief summary of the whole report.
    b. Description of implementation. How did you implement your class? We don't want to read a bunch of code, but if some pseudo-code helps you explain what you did, you are welcome to use it.
    c. Report on the results of following your test plan. You must provide evidence that in carrying out your test plan that your class has (or has not) passed your tests.
    d. Acknowledgments. You must acknowledge all help received. You must cite and quote all sources of material that you did not create yourself.
    e. Help Given. Please explain any help that you gave to other students.
    f. Your code. Provide a printout from Eclipse of your code including line numbers. Be sure to print out the final version and not some version that does not match what you will upload to Learn. Please do not print from Notepad. Please do not copy and paste into MS-Word. Print from Eclipse.

What to hand in for both parts (1 and 2):

1. You should copy and rename your program files (the **.java** files, often named something like Program.java) to be nexususername-proj1-partN-Filename.java where nexususername is your nexus username and partN is either part1 or part2, and Filename.java is the original filename of the code, and **upload it to the UW-Learn** drop box for Project 1 - Part N. Do not make mistakes naming your file. We will compile your program and run it.
2. All pages of your report must be letter-sized pieces of paper. You must staple multiple pages together with one metal staple in the upper left hand corner. Your stapling should be neat and tidy. Mangled staples are not permitted. Do not use report binders. You are welcome to use duplex printing to save paper.
3. You should hand in the paper portion of the assignment to the dropbox on the ~~second~~ **third** floor between E2 and CPH.

Failure to upload your file correctly will result in a significant loss of points. The timestamp that UW-Learn gives your upload will determine when you have turned in the homework, but your hardcopy version must also be placed in the dropbox on time.

Marking Rubric

- Each report is worth 50 pts. These points will be divided as follows:
    - Part 1:
        - Class design (public interface): 10 pts
        - Quality of test plan and code: 30 pts
        - Overall quality of report and the remaining sections: 10 pts. These points include using good style for your Java code and a neat and professional report.
    - Part 2:
        - Class implementation: 25 pts
        - Quality of report of test plan results: 15 pts
        - Overall quality of report and the remaining sections: 10 pts. These points include using good style for your Java code and a neat and professional report.
- Please note that there are lots of points to be earned by turning in a report. Just because you are having some problems with your program, does not mean you cannot write a report! Turn in a report!

Collaboration and Academic Honesty

1. Please review the syllabus for guidelines and extensive rules on academic honesty in this course.
2. You are to work alone on this assignment except that:
    a. You may talk with the instructor and the TA regarding this homework.
    b. You may talk with students about general programming questions, but you may not discuss ways to solve this homework. For example, you may have another student help explain loops or conditionals or variables to you, but you may not discuss how to write parts of a program to solve this homework.
3. All collaboration other than that allowed above (2a, 2b) is forbidden. For example:
    a. You may not discuss the problems with people.
    b. You may not sketch out pseudo-code on a whiteboard.
    c. You may not share code with others.
    d. You may not give answers to others.
    e. You may not compare answers with others.
4. In addition, you may not use any materials other than your textbook, books you can find in the library, and Java API documentation. You must cite any source that you use. **You may not search for answers on the internet.**
5. We will use automatic methods to detect cheating as well as human inspection of solutions.
6. Violating these rules will be considered academic misconduct and subject to penalties.