**Project Two**

**Zachary Carper**

**CS-320-T5618**

**17 June 2022**

For this project the approach I took seemed to work. Honestly, I’m glad I’ve done Java a great amount before taking this class and I felt like I wasn’t improving as I should, but it has shown throughout this class that I’m doing better than I expected. I had a rewrite and fix my code multiple times as I’m still having some issues mainly with still trying to figure out some of this compatibility with the M1 architecture. I had a few minor issues with the development of the code in the actual classes, but I found some online tools to assist me in that for some problems I was facing as well as some prior notes from other classes as most of my issues usually are me overthinking things and making it harder than what it really is. I think my Junit tests would have succeeded; they did for me at least.

I believe most of my code was great and met the rubric requirements. For trying to configure how everything was just laid out was the hardest thing for me to accomplish. I always struggle with how it is laid out; I credit that to watching YouTubers who have been coding for 15 years and work at Google and setting the expectation to make it as neat, clean, and functional as they do. For my test I thought I wrote it pretty clean and simple, and it functioned just fine when I ran it:

**public** **static** **boolean** addTask(task task) {

**boolean** alreadyTask = **false**;

**for** (task taskList : *tasks*) {

**if** (taskList.getID().equals(task.getID())) {

alreadyTask = **true**;

}

}

**if** (!alreadyTask) {

*tasks*.add(task);

}

**return** alreadyTask;

}

**public** **static** **boolean** deleteTask(task task) {

**for** (task taskList : *tasks*) {

**if** (taskList.getID().equals(task.getID())) {

*tasks*.remove(task);

**return** **true**;

The test ran like I thought they should, and they worked properly. My main goal in this project and all others is to just follow the rubric and do as it say. This project took time but I’m thankful we broke it up over different modules because that would have been an insane overload in one week to do so. I believe in my next classes I’ll be expecting timelines such as that so I’m glad I’ve been going through the repetitions now in this class and that I’ve had the proper time to address this to fix those issues now. I do feel I need to utilizing comments more and have more thorough explanations on my code, for example this code below:

**public** Contact(String contactId,String fName,String lName,String phone,String address){

**if**(contactId.length() <= 10 && contactId != **null**) {

**this**.contactID = contactId;

}

**this**.setFirstName(fName);

**this**.setLastName(lName);

**this**.setPhoneNumber(phone);

**this**.setAddress(address);

}

**public** **void** setFirstName(String fName) {

**if**(fName.length() <= 10 && fName != **null**) {

**this**.firstName = fName;

}

}

**public** **void** setLastName(String lName) {

**if**(lName.length() <= 10 && lName != **null**) {

**this**.lastName = lName;

}

}

**public** **void** setPhoneNumber(String phoneNumber) {

**if**(phoneNumber.length() == 10 && phoneNumber != **null**) {

**this**.phoneNumber = phoneNumber;

}

}

**public** **void** setAddress(String address) {

**if**(address.length() <= 30 && address != **null**) {

**this**.address = address;

}

}

**public** String getContactID() {

**return** contactID;

I feel when writing code such as this and looking back on it now I could have utilized comments more efficiently and utilized more detail when explaining everything I’ve done. It’s hard for me realize that someone else may be reading this and they may have a more difficult time following along. Maybe they wouldn’t have an issue, but I think utilizing more comments on my code would be more professional and negate the chance of that happening. Next time I think I’ll slow down the pace in which I try to accomplish these projects and treat it more like a real occupation because one day it will be. I felt I definitely cut corners on my explanation of this and that is an improvement I need to work on in the future. In order to keep myself in check when developing this I did what I do at work and that is by finishing my code for a certain portion on for example Monday and then coming back on a Tuesday and looking at it from a fresh view. I think critiquing your own work after immediately creating it can make us easily look over errors and not notice them. I’ve always done this with everything in school such as writing papers. I started this in high school, and it has proven to be successful. This is how I ensure I don’t cut corners but there is still progression I can make and will always be able to make. In the past when not readdressing my work because I was rushing I’ve made major mistakes in my real occupations that could have had dire consequences if one of my coworkers wouldn’t have caught it.