Term Portfolio Project

Eric Lin

V01001632

Seng265: Software Development Methods

I. Introduction

My learning jouney has always been emerged in softwares, and utilizing different tools to its maximum potential. As a student in the 21st century, it is obvious that software development will be leading of the future, establishing its importance in today's society. Personally, one great example would be desmos which was heavily used in my highschool for math visualization, and it was immensly helpful for such software to exist in my generation. I would like to use the oppurtunity of writing this document to record my experience in software development methods. This document will consists my weekly experiences with Jupyter notebook, Unix/Linux, git, github, and gitlab, as well as a couple coding languages, C and Python. The document will separated into 2 parts, with part 1 consisting mostly about Jupyter notebook, and part 2 mostly about git, github, gitlab, and my experiences with C and python. The 2 parts of the documents will show my knowledge with different software development methods as part of my learning journey.

For part 1, it will include the following sub-topics:

- The **core functionality** of Jupyter Notebook
- Simple method of editing **Jupyter Notebook markdown** including insert links and images
- **Typeset mathematical formulas** using LaTeX markdown in Jupyter Notebook
- The **history of Unix** and how prevalent is it in software development today

For part 2, it will include the following sub-topics:

- The usefulness of **SENG 265 for building my resume** for future job applications
- **Most valuable skills from SENG 265 and skills I am adding to my resume**
- The notion of and **motivation for typing and typing hints in Python**

- The Python concepts of **slicing** and **comprehension**

- The importance of **implicit or tacit knowledge** in **requirements engineering** and **job interview dialogues**

- The concepts of **deep copy** and **shallow copy** of objects as well as the purpose of the methods **__copy__()** and **__deepcopy__()**

- **How challenging was Assignment 3 for me**

- **My continued learning experience in SENG 265**

In the end, this document also serves an important meaning in my learning journey, as it labels down important milstones of my study in software development methods. In today's society, the ability of acknowledging the usefulness of the tools around you and using it is highly valued. I hope this would be a good demonstation of myself capitalizing the tools around me for future success. Especially, this document should reflect on the progress of my learning journey and improvement in my future skillset.

II. The core functionality of Jupyter Notebook

Jupyter Notebook is a extremely powerful tool for creating projects. As programming has been a major part of presenting large datas in pioneering compnaies, being able to integrate codes into projects is needed for a more efficient and understandable workflow. The "notebook" of Jupyter notebook represents the idea of utilizing codes into presenting datas, charts, and other useful medias. Notebook also provides projects with another major aspect, shareability. Since Jupyter notebook is provided as an open source, everyone has access to it, which means being able to share and edit notebooks plays a important role when sharing data in a readable form.

There are four components in Jupyter Notebook:

1. Notebook Document
    - Notebook documents (or “notebooks”, all lower case) are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc…). (What Is the Jupyter Notebook? — Jupyter/IPython Notebook Quick Start Guide 0.1 Documentation, n.d.)

2. Jupyter Notebook App
    - The Jupyter Notebook App is a server/client application which can be installed on local desktop or a server. If the app was used on a local desktop, it requires no internet access and is used through a web broweser. (What Is the Jupyter Notebook? — Jupyter/IPython Notebook Quick Start Guide 0.1 Documentation, n.d.).

3. Kernel
   - A notebook kernel is oftern refered as a computational engine that executes the code in a notebook document. Kernel usually runs with the whole document instead of only for a specific cell in a notebook document. Jupyter Notebook provides different functions for kernels including Restart, Restart & Clear, and Restart & Run All, which all serves it's own puposes when kernel needs to be interrupted manully. There are also many different kinds of kernels. Although the one being used in Jupyter Notebook is for python codes, there different kernels for different versions of python and also for up to 100 different other languages (Pryke, 2023).

4. Notebook Dashboard
   - The dashboard is the interface first shown in the Jupyter Notebook App. It is used for navigating through local files, like a file explorer.

Personally, this is my first document written in Jupyter Notebook. Although I have nearly no experience with Jupyter Notebook, I can see the potential it holds when there is data needed to be presented. Especially, Jupyter Notebook can come in very handy when the data is used with codes. When there are more opputunity in the future, I will definitely consider Jupyter Notebook as one of my options for making data related projects.

III. Simple method of editing Jupyter Notebook markdown

Editing Jupyter Notebook markdown is simple and does not require much time to learn as if it was another coding language. It is very similar when compared with HTML tags. A quick exmaple will be something like this:

`This is when writing normal text`

`# This is a level 1 heading`

`## This is a level 2 heading`

`[This is how you include hyperlinks](https://www.youtube.com)`

`This is how you add images: ![Alt text](https://www.example.com/image.jpg)`

The output should look like this:

This is when writing normal text

# This is a level 1 heading

This is a level 2 heading

This is how you include hyperlinks

This is how you add images: Alt text

(Since I did not put in a valid link, there was no image shown)

There are also other ways to include an image in your notebook document (Pryke, 2023).

- Use a URL from the web for an image
- Use a URL from your local files for an image that you will keep beside with your notebook document
- Use the Edit tab in Jupyter Notebook to insert an image in your file. (Your .ipynb file will become much larger)

IV. Typeset mathematical formulas using LaTeX markdown in Jupyter Notebook

Typesetting mathematical formulas in Jupyter Notebook is by utilizing the already exisisting typesetting method, LaTeX. The command used in Jupyter Notebook is no different than directly typing mathematical formula in LaTeX. The Jupyter Notebook uses MathJax to render LaTeX inside HTML / Markdown. Just put your LaTeX math inside `$ $`. Or enter in display math mode by writing between `$$ $$` (Aziz, 2021). For example:

This is how you type Euler's identity: `e^{i \pi} + 1 = 0`

If wrapped around inside the , it sould look like this: $e^{i\pi} + 1 = 0$

And it should look like this when entered in display math mode,

Euler's Identity:

$$e^{i\pi} + 1 = 0$$

There are many more to typesetting in LaTeX such as using `\\` for a new line, `\frac{arg1}{arg2}` for fractions, `^{}` for powers, and `_{}` for indicies, and many more. Furthermore, there are also greek letters in LaTeX which are usually used with backslash and the letter name. For exmaple `\delta` δ, `\omega` ω, or `\pi` π

I have not had many experiences with typesetting mathematical formulas in Jupyter Notebook Markdown. However, I am curretnly taking the course MATH122, which makes turning in assignments optional with LaTeX. I have been utilizing that opputunity to the fullest and are constantly learning new methods in typesetting better in LaTeX. At the start I was struggling with labeling my questions, and now I can properly align my formulas in a organized manner with desired symbols. In the future, I will surely use more of LaTeX in providing my documents a better way of showing mathematical formulas.

V. The usefulness of SENG 265 for building my resume for future job applications

Throughout the course of SENG 265 I have completed many different achivements, for example, learning the programming languages C and Python, studying the functionality of bash and vim, and researching about the history of Unix/Linux and LaTeX. There are many more to talk about, but the listed achivements has been the major contributions to the skillset I have today.

Aside from the hard skills, one of the main goals I will soon copmlete in SENG 265 is improving my tacit knowledge, in terms of software developing. According to Cambridge Dictionary, Tacit Knowledge is "the knowledge that you do not get from being taught, or from books, etc. but get from personal experience, for example when working in a particular organization" (Tacit Knowledge, 2023). In many assignments, there consist multiple different ways to approach the same question, however, it all comes down to the student to decide how are they going to approach it. Thus, the tacit knowledge is important for the student to find out what could potentially improve the efficiency of the program. Personally, I always do a great deal of researching when doing any of the assignments, hoping to achieve the best possible approach. Therefore, I would consider the amount of tacit knowledge from the countless hours I have spent on each assignments was quite optimal.

In the end, SENG 265 not only increased the skillsets I can put on my resume, it also helped me in many areas I was weak at before taking the course. It was a jouney to work around each obstacles for the assignments, labs, and in-class practices. I am sure this course is the most helpful course in terms of resume building. I would also consider myself decently confident in the future interviews.

VI. Most valuable skills from SENG 265 and skills I am adding to my resume

The course SENG 265 covered plenty of skills that is notable for any software developers. Consisting the following:

- Programming Languages
  - C
  - Python
- Bash
- Vim
- Unix/Linux
- Static and Dynamic Data Structures
- Software Development Cycle
- Version Control System (VCS)
- Git
- Static and Dynamic Data Structures

- Object Oriented Programming (OOP)

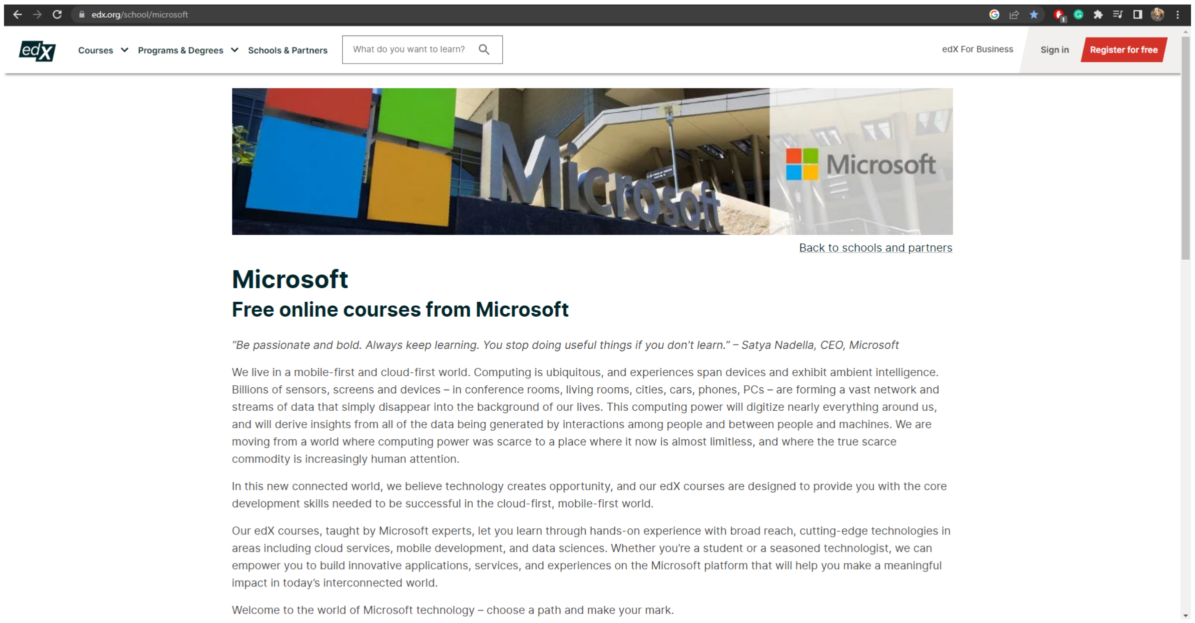
Although SENG 265 covers a wide range of skills and was taught thoroughly with each of them. However, there was only some of the skills I could fully attain and which I would likely to put on my resume. Those skills are C, Python, Bash, and Static and Dynamic Data Structures. Despite I am not so familiar with the other skills, or even some of the ones I might add on my resume, but I am certain I will take time to study and make sure they meet a high standard in order to become my own skills.

In my opinion, the most valuable skills taught in SENG 265 are the programming languages C and Python. These two programming languages are both in the top 3 programming languages in the world (TIOBE Index - TIOBE, 2022). It is most likely the first job of any software developers would be using one of those programming langauges in the top 3. Moreover, having a wide range of programming languages are also incredibly important when any software developers are trying to enter the work field. By being fluent in many languages, it can be regarded as your personal versatility when working with different environments, no matter if it is coding or in real life working environment. As it might be more competitive then one would think, having more programming languages in your skillset really makes a difference.

My plan in the near future is to review the basic concepts of C and Python. Since I am still a first year student in computer science, programming in SENG 265 was not the ideal environment for me to smoothly learn the language. As I have mentioned above, I always do plenty of research before and during any assignments, part of the reason being I am not as smooth with coding in either languages. Therefore, during the summer, I plan on learning C from Professor Bill Bird's lectures (BillBird, 2021) on YouTube and python on edx.org (Microsoft's free online courses) (edX, n.d.). I am fairly confident with the end of summer, I will be able to code in C and Python as fluent as someone who took Python and C regularly.

The following are screenshots of the websites discussed above:

- EDX



Microsoft

Free online courses from Microsoft

"Be passionate and bold. Always keep learning. You stop doing useful things if you don't learn." – Satya Nadella, CEO, Microsoft

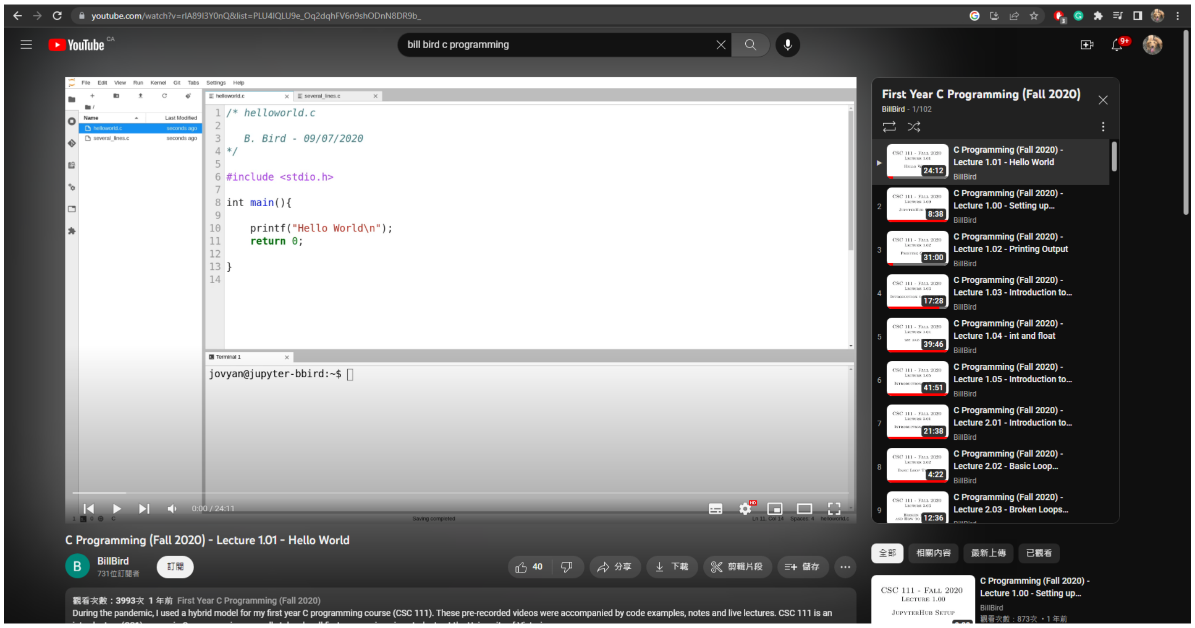
We live in a mobile-first and cloud-first world. Computing is ubiquitous, and experiences span devices and exhibit ambient intelligence. Billions of sensors, screens and devices – in conference rooms, living rooms, cities, cars, phones, PCs – are forming a vast network and streams of data that simply disappear into the background of our lives. This computing power will digitize nearly everything around us, and will derive insights from all of the data being generated by interactions among people and between people and machines. We are moving from a world where computing power was scarce to a place where it now is almost limitless, and where the true scarce commodity is increasingly human attention.

In this new connected world, we believe technology creates opportunity, and our edX courses are designed to provide you with the core development skills needed to be successful in the cloud-first, mobile-first world.

Our edX courses, taught by Microsoft experts, let you learn through hands-on experience with broad reach, cutting-edge technologies in areas including cloud services, mobile development, and data sciences. Whether you're a student or a seasoned technologist, we can empower you to build innovative applications, services, and experiences on the Microsoft platform that will help you make a meaningful impact in today's interconnected world.

Welcome to the world of Microsoft technology – choose a path and make your mark.

- Professor Bill Bird's C Programming Video



```c
/* helloworld.c

   B. Bird - 09/07/2020
*/

#include <stdio.h>

int main(){

    printf("Hello World\n");
    return 0;

}
```

```
jovyan@jupyter-bbird:~$
```

C Programming (Fall 2020) - Lecture 1.01 - Hello World

觀看次數：3993次 1 年前 First Year C Programming (Fall 2020)
During the pandemic, I used a hybrid model for my first year C programming course (CSC 111). These pre-recorded videos were accompanied by code examples, notes and live lectures. CSC 111 is an

First Year C Programming (Fall 2020)

1. C Programming (Fall 2020) - Lecture 1.01 - Hello World
2. C Programming (Fall 2020) - Lecture 1.00 - Setting up...
3. C Programming (Fall 2020) - Lecture 1.02 - Printing Output
4. C Programming (Fall 2020) - Lecture 1.03 - Introduction to...
5. C Programming (Fall 2020) - Lecture 1.04 - int and float
6. C Programming (Fall 2020) - Lecture 1.05 - Introduction to...
7. C Programming (Fall 2020) - Lecture 2.01 - Introduction to...
8. C Programming (Fall 2020) - Lecture 2.02 - Basic Loop...
9. C Programming (Fall 2020) - Lecture 2.03 - Broken Loops...

VII. The notion of and motivation for typing and typing hints in Python

Type hints are useful tools to label the types in the programming languages Python to clarify the type of variables, functions, or classes that does not have type hints. Here is a general expression of type hints for variables and functions:

```
song: data type = "Smell Like Teen Spirit"
```

```
def greet(): -> data type
    print("Hello world!")
```

Here are some examples:

`song: str = "Smell Like Teen Spirit"`

```
def greet(): -> None
    print("Hello world!")
```

One detail worth noting is when nothing is return from a function, it return "None".

In general, the type hints for variables are typed by adding a colon and the data type of the variable. The type hints for functions are added by adding an arrow "->" after the colon of the function.

Since typing is not enforced in Python, it is not neccessary in order for the program to run successfully. Some might ask "why should we use type hints if it is not enforeced by the intepreter?" There are some codebases that are large and hard to code trace, and type hints in those kinds of circumstances would be very helpful. Additionally, there are tools such as mypy which enforces the type hints in python. This is very useful for software developers to detect minor errors spread out in the codes. It could prevent hours of debugging for minor mistakes that probably does not contribute to any logical malfunctions of the program. Therefore, type hints can also be helpful when used with type hints enforcing tools.

As a beginner in Python, I have not fully experienced the benefits of type hints due to most of my projects still remain small in size. In the future, I believe there will be a great deal of opputunity for me to experience type hints. At that time, I should be able to see the all the benefits of type hints, and potentially utilize it as I code.

VIII. The Python concepts of slicing and comprehension

List comprehension lets programmers utilize the exisisting values in a list to make a new list with shorter syntax. As we would usually use a for loop to make a new list, to loop through the original list to attain values. We can see examples from the W3Schools (Python - List Comprehension, n.d.):

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []
```

```
for x in fruits:
  if "a" in x:
    newlist.append(x)
```

```
print(newlist)
```

List comprehension lets us achieve the same result using shorter syntaxes.

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
```

`newlist = [x for x in fruits if "a" in x]`

`print(newlist)`

The general syntax for list comprehension:

`newlist = [expression for item in iterable if condition == True]`

List comprehension gives a better way of writing a common code that programmers often encounters, making the code more readable. This reduces the chance of codes clogging up or perhaps become lengthy and hard to debug. List comprehension is also a great way to loop through a list multiple times to filter the items for the wanted result.

List slicing is useful when trying to get certain values inside a list. The general syntax for slicing is the following (H, 2022):

`my_list[start:stop]`

where start is the index of the first element to include, and stop is the index of the item to stop at without including it in the slice (H, 2022).

If we define my_list as `my_list = [_ for _ in '123456']`,

`my_list[0] returns '1'`

`my_list[0:3] returns ['1','2','3']`

If you leave the boundaries blank, then it either means to start from or to the end of the list.

`my_list[3:] returns ['4','5','6']`

`my_list[:3] returns ['1','2','3']`

Quite interestingly, slicing in Python also accepts negative boudaries. The index of the negative boudaries goes as: the right most (the last) item in the list is -1, and the index decrement as it approches the left (the start) of the list. For exmaple:

`my_list[-3:-1] returns ['4','5']`

A very important note to take here is when dealing with negative boundaries, if the starting index is a smaller number than the stopping index, then an exmpty list will be returned such as `my_list[-1: -3]`. This is caused by the boundaries not meeting the slicing requirement, which is the starting index should be at the left of the stopping index and there should be elements on or to the right of the starting index. However, `my_list[-1:-3]` has its starting index of the right of the stopping index, so and empty list is returned before anything is sliced.

One other slicing technique is "stepping", it acts like if you are skpping elements in a list. The negative indexes still applies here, but it returns the value in reversed order. Some example of stepping are (H, 2022):

`my_list[::2] returns ['1','3','5']`

`my_list[1::2] returns ['2','4','6']` (start from index 1)

`my[::-1] return ['6','5','4','3','2','1']`

`my_list[-1:-5:-2] returns ['6','4','2']`

In my experience of coding with Python, I have not had many chances to use either list comprehension or slicing. There was only a little bit of slicing included in my assignment 2. However, I can see how this is very useful for someone to retrive values from lists. For list comprehension, it can be tremendously helpful if codes were reduced and made the codebase more concise. In general, these two techniques would definitely come in handy in the future, no matter if it is for my co-op job or school projects, and I will take advantage of utilizing this skill.

IX. The importance of implicit or tacit knowledge in requirements engineering and job interview

Tacit knowledge, different from explicit knowledge, is the often times known as the knowledge you gain from living. Tis kind of knowledge is informal and hard to convey or express. Tacit knowledge is hardly attained from reading text-book, manuals, or watching videos. It is usually knowledges developed from the experiences you have throughout your life, such as:

- Leadership
- Innovation
- Commitment

These knowledges are often regarded as subjective, as it comes out differently from person to person since everybody has different experiences. A more detailed explanation can be found in "What is Tacit Knowledge: Importance, Benefits & Examples", as it lists tacit knowledge is often expressed in (Prabhakaran, 2023b):

- Behaviours
- Actions
- Habits
- Responses
- Intuitions

Moreover, tacit knowledge are usually developed from (Prabhakaran, 2023b):

- Trials and Errors
- Experiments
- Capturing data throughout a research period
- Documenting findings, then using the information to strategize

During a couple lectures in SENG 265, we discussed the importance of tacit knowledge in interviews. Professor Muller used an interview question exmaple from Google to imitated the environment when we get interviewed. The results of this simulation was to convey the message of knowing the most explicit knowledge is not the most important when doing an interview, the process of approaching an answer is what matters the most.

In an engineering setting, when interviewers watch the person using a known method to find the result, the interviewer could only evaluate the person's skill level which everybody has. However, if a person shows interests in the question and is eager to solve it by asking questions, thinking out loud, and shows innovation; even if the interviewee could not solve the question, the interviewer would likely learned more about this person than the former. That is a perfect example of tacit knowledge.

In the end, tacit knowledge includes many personal triats which can indicate a person's success potential in a organizational setting. It would be the most optimal if the person can make contributions to the company.

Personally, I have not had many opportunities to demonstrate my tacit knowledge. Perhaps the first time I could demonstrate my tacit knowledge would be my first interview. However, I will make sure I am accumulating it throughout my university jouney, in order for me to be the best of myself when being interviewed in the future.

X. The concepts of deep copy and shallow copy of objects as well as the purpose of the methods __copy__() and __deepcopy__()

Deep copy and shallow copy act as two different functions in Python. The two methods of copy should be used according to the situation and the purpose of the code. In short, shallow copy only copies the first layer of a compound object —list, dicts, and sets— and not the properties inside the object. Deep copy will copy the compound object and also the properties inside the object.

The syntax for shallow copy is (Copy — Shallow and Deep Copy Operations, n.d.): `copy.copy(object)`

ex. `groups = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`

```
`new_groups = copy.copy(groups)`
```

The syntax for deep copy is:

`copy.deepcopy(object)`

ex. `groups = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`

```
`new_groups = copy.deepcopy(groups)`
```

Using the example above, the shallow copy function will only copy the `groups` over to the `new_group`, however, it will not copy the items inside the list, the items will be references of the original object's items. Therefore, the id of the object will be different, but the id of the items will be the same, when comparing groups and new_groups.

On the other hand, the deep copy function will copy both the object and the items of the object over to the new object. Therefore, the object will have different ids, and so do the items within both objects.

`copy()` is especially usefull when one wants to change the data of a compound object, but does not want to change the original object's values. `copy()` then comes in handy to make a replicated version of the original object, and the user can change its value without chaning the original object's value. For expample, using one of the most used data structure in assignment 2, pandas. If the user wants to copy a Data Frame, using `copy()` would be extremely useful to make sure the original values are kept for future purposes.

The pupose of the methods `__copy()__()` and `__deepcopy__()` act as a tool when one wants to implement their own implementation of shallow copy and deep copy in a class. The former is shallow copy. The latter is deep copy.

XI. How challenging was Assignment 3 for me

As a first year undergraduate computer science student, I consider assignment 3 as one of the hardest assignment I have ever done in my programming career (close to 500 lines of codes + comments). Since it has been my first experience coding in C in SENG 265, I was really unfamiliar with many errors in C. Previously I have learned Java for a decent amount of time, and I consider myself fluent in Java. However, C was only similar to Java when compared with other programming languages like Python.

At the start of the assignment I was grateful for my past self to study about C in the first assignment, as the knowledge was transfered to assignment 3 and got me a decent start on how to approach the assignment. Nevertheless, I started to struggle when trying understand the concept of linked list in C. It was taught in my last year's CSC 115 class, but again, it was in Java. There are also many elements in C that are completely different when compared with Java such as interfaces, private variables, nodes, or file functions. Perhaps it would be better if it was mentioned in class for better understanding for Java-fluent student, but considering I was probably the minority, I decided it was my responsibility to make sure I allocate more time than others to study the language thoroughly.

During my coding sessions for assignment 3, it was dreadful when dealing with memory errors such as "aborted (core dumped)" or "Segmentation Fault". I asked many of my friends for help and was able to deal with the errors one by one. Eventually, I was somewhat efficient at finding segmentation faults source, and could debug memory errors at a decent speed.

In conclusion, I consider assignment 3 as a very good practice for someone who would like to get more familiar with memory allocation in C. It might not be a good assignment for someone who just started learning C. I was fortunate to have a head start on getting familiar with C as a language before tackling this assingment, so all I had to deal with and learn from is memory issues.

XII. My continued learning experience in SENG 265

So far, SENG 265 has been one of the most challenging course I have ever taken in my programming career. There are plenty of knowledge I have acquired from SENG 265, and I believe majority of the knowledge are resume-worthy. In general, SENG 265 has been a challenging but incredibly productive class for anyone's career in terms of being a software developer.

As we learned about bash at the started of the term, we explored a way to renew our versions of code when each code session ends. This was extremely helpful when trying to work in a remote setting where codes are needed to be transfered beteen devices. I have been utilizing this funtion to code in different settings. We then learned about C and Python, which is heavily related to the working field we have today in order to become a programmer. C and Python are considered to both be in the top 3 programming languages in the world, which determined the usefulness of SENG 265 in a statistical way. For me, this was a immense achievement as a computer science student, where learning new languages and become fluent should be a milstone.

By combining everything we have learned in SENG 265, we are able to become a better software developer with the increased skillset. Personally, I would keep studying the materials given in SENG 265. Since I have not been too familiar with the functionality of many things such as git, vim, bash, unix, and other VCSs, studying them would be my next goal. I believe it will be tramendously helpful when going into any automation related work setting such as developing robotics, artifical intelligence, or even simple program automations.

Studying the given mateirial is one part, I will also try to do programming practices on LeetCode, watch videos of tips in interviews, and study the mathematical theories in my future classes. There are much to do to become a profoud software developer. I will keep learning and develop my tacit knowledge to hopefully be a successful software developer.

XIII. References

What is the Jupyter Notebook? — Jupyter/IPython Notebook Quick Start Guide 0.1 documentation. (n.d.). https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#kernel

Pryke, B. (2023, February 8). How to Use Jupyter Notebook: A Beginner’s Tutorial. Dataquest. https://www.dataquest.io/blog/jupyter-notebook-tutorial/

Math and equations. (n.d.). https://jupyterbook.org/en/stable/content/math.html#equation-my-label

Aziz, K. A. (2021, December 14). Learn How to Write Markdown & LaTeX in The Jupyter Notebook. Medium. https://towardsdatascience.com/write-markdown-latex-in-the-jupyter-notebook-10985edb91fd

Basic Syntax | Markdown Guide. (n.d.). https://www.markdownguide.org/basic-syntax/

tacit knowledge. (2023). https://dictionary.cambridge.org/dictionary/english/tacit-knowledge

TIOBE Index - TIOBE. (2022, June 3). TIOBE. https://www.tiobe.com/tiobe-index/

edX. (n.d.). Microsoft. https://www.edx.org/school/microsoft

BillBird. (2021, July 18). C Programming (Fall 2020) - Lecture 1.01 - Hello World [Video]. YouTube. https://www.youtube.com/watch?v=rlA89I3Y0nQ

Radečić, D. (2022, March 23). Type Hints in Python — Everything You Need To Know In 5 Minutes. Medium. https://towardsdatascience.com/type-hints-in-python-everything-you-need-to-know-in-5-minutes-24e0bad06d0b#:~:text=As%20the%20code%20base%20gets,you're%20using%20type%20hints.

Besbes, A. (2023, February 10). 12 Beginner Concepts About Type Hints To Improve Your Python Code. Medium. https://towardsdatascience.com/12-beginner-concepts-about-type-hints-to-improve-your-python-code-90f1ba0ac49#:~:text=Type%20hints%20are%20performed%20using,not%20enforced%20on%20the

Python - List Comprehension. (n.d.). https://www.w3schools.com/python/python_lists_comprehension.asp

GeeksforGeeks. (2023, February 19). Python List Comprehension and Slicing. https://www.geeksforgeeks.org/python-list-comprehension-and-slicing/

Ramakrishnan, M. (2023, January 24). What is List Comprehension in Python and its Advantages? Emeritus Online Courses. https://emeritus.org/blog/coding-what-is-list-comprehension-in-python/#:~:text=You%20can%20use%20list%20comprehension%20in%20Python%20to%20repeat%

H, J. (2022, March 30). The Basics of Indexing and Slicing Python Lists - Towards Data Science. Medium. https://towardsdatascience.com/the-basics-of-indexing-and-slicing-python-lists-2d12c90a94cf

Prabhakaran, J. (2023b, January 23). What is Tacit Knowledge: Importance, Benefits & Examples. Document360. https://document360.com/blog/tacit-knowledge/#:~:text=Tacit%20knowledge%20is%20the%20knowledge,our%20personal%20beliefs%2

Jalli, A. (2022, January 6). Shallow Copy vs. Deep Copy in Python - Better Programming. Medium. https://betterprogramming.pub/shallow-copy-vs-deep-copy-in-python-357e5f502bf9

copy — Shallow and deep copy operations. (n.d.). Python Documentation.
https://docs.python.org/3/library/copy.html#:~:text=A%20shallow%20copy%20constructs%20a,obje

Python, R. (2023, April 1). Shallow vs Deep Copying of Python Objects.
https://realpython.com/copying-python-objects/