**Wayne State University**

**Upload Date: Feb 8, 2024**

**CSC 4110 - Software Engineering**

**Weekly Homework**

**Video Link:** [**https://www.youtube.com/watch?v=-t9MotIkS-k**](https://www.youtube.com/watch?v=-t9MotIkS-k)

**Directions:**

**Do all problems by the due date. Follow instructions explicitly. See general requirements for each problem.**

**There are TWO parts to this assignment:**

**CODING and QUESTIONS/ DOCUMENTATION**

**Part One: CODING**

See **General Requirements (pg 5)** - some **Problems have ‘Specific’ requirements**

**Directions:**

* Customer requests to be fulfilled **ON-TIME**
* .py as .txt file to be uploaded to GitHub repository (see below links)
* GitHub link placed in COMMENT section of Upload folder
* Code images and output images placed in THIS original assignment, uploaded to course shell (with appropriate comments, etc…)
* Comments appropriate and explanatory, contain all necessary information

**Problem One**

**\*\*\*\* Adhere to ‘General Requirements’**

**Assignment: Non-disparate DATA WAREHOUSING**

Simulate non-disparate data warehousing with the following **sequence**:

**Step One: ‘data collector’**

Create a **data collector** method, simulating user records with the following attributes: username, password, birthdate, address, social security number,productPurchased,salesperson. This program **procedurally generates ‘sample’ data and stores that data.**

**Step Two: ‘key/value’ pairs**

Feed data collector values into ***key-value pair***. For example, the user data may be an entire list sequence, which is then considered the ‘value’ portion of a unique user ID key.

**Step Three: search engine**

This key-value structure must be searchable. For example, a user may be able to search the entire data store for users in a certain state, or see which users were handled by a certain salesperson (or sales ID).

**Problem One Requirements**

“ProductPurchased” consists of order/vendor information such as usernames, web orders, product IDs, quantities, date of order, region, etc.. Any item referring to products, such as Product ID should have the prefix “ID,” such as “ID-trxdfn.”

*The specific design, method and procedure details are up to the student; the column/ category names are up to the student.*

***The student MUST create a DATA COLLECTOR method that PROCEDURALLY generated USER DATA; then the DATA must be placed into key/value pairs and be searchable.***

**A screen shot of a computer

Description automatically generatedA screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**ProblemTwo**

**\*\*\*\* Adhere to ‘General Requirements - LAST PAGE’**

**Assignment: Create a ‘Game of Chance’**

Create a ‘game of chance’ simulation to do the following:

(a) build and populate treasure chest with as many items customer requires

(b) create a bank / loot stash

(c) wagers to be placed per “spin” or treasure chest “grab”

(d) customer “plays” until bank account reaches 0 or below.

**Problem Two Requirements:**

Note: the name of the simulation shall be “pirate” related; copy/ paste code and output, showing different outcomes; “**random**” module is to be imported.

A screen shot of a computer program

Description automatically generated

**ProblemThree**

**\*\*\*\* Adhere to ‘General Requirements - LAST PAGE’**

**Assignment: Create password simulator**

Customer needs a password simulator to do the following:

(a) create random passwords in perpetuity

(b) if the password is “acceptable,” it gets archived

(c) “unaccepted” passwords get deleted

(d) no less than 40 iterations

**Problem Three Requirements:**

Customer rules of ‘accepted passwords’ include: “special symbols,” and password cannot be a word in a dictionary list; “**random**” module to be imported.

A screen shot of a computer program

Description automatically generated

**General Requirements:**

1. **Add labeling/ comments (name, date, revision #); add in-line requirements where appropriate (such as syntax usage).**

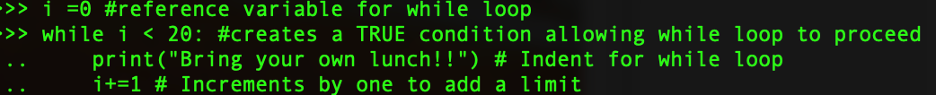
1. **AT LEAST ONE PROBLEM MUST USE SONIFICATION AND VISUALIZATION.**

#Indicate coding begin and end

Example acceptable code comment:

**# Revision number BEGIN/ START DATE**

**## Begin John D. Student here (date)**



**# Revision number FINAL DATE**

**## End John D. Student here**

**# Group / manager/ lead tech/ project # ←-Where appropriate**

1. **Adhere to the following coding style (from PEP8):**

1. Wrap lines so that they don’t exceed 79 characters.

2. Use blank lines to separate functions and classes, and larger blocks of code inside functions

3. When possible, put comments on a line of their own.

4. Where appropriate, name your classes and functions consistently; the convention is to use UpperCamelCase for classes and lowercase\_with\_underscores for functions and methods.

1. **GitHub:**

GitHub Video 1: <https://www.youtube.com/watch?v=fJtyf62yAb8>

GitHub Video 2:<https://www.youtube.com/watch?v=GqNAD4XoZ6k>

**Reference following article to create repository so you can load this assignment output:**

[**https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop**](https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop)

1. **Fill out below CRD**

**Change Request Document**

**Name: Talal Hussain**

**Student access ID: HI5121**

**Project: Homework 4**

**Date: February 15, 2024**

ProductPurchased Change

**1.** **Change Request and concepts*:***

Within problem 1 I added the product ID generator. This will help keep track of the product id’s and show what the customer bought with the vendor information and dates.

***3.*** ***Highlighted Source Code:***

product\_purchased = {}

for l in range(20):

product\_purchased["ID-" + str(l)] = {

"OrderID": "Order" + str(l),

"Vendor": random.choice(usernames),

"ProductID": "ID-" + str(random.randint(0, 19)),

"Quantity": random.randint(1, 10),

"DateOfOrder": random.randint(1, 31),

"Region": random.choice(["North", "South", "East", "West"])

}

**Part Two: QUESTIONS**

**Answer End of Chapter Questions with Real-Life Examples, documented by APA references, at least ONE reference per Question.**

See following link to automatically CREATE your references:

<https://owl.purdue.edu/owl/research_and_citation/apa_style/apa_formatting_and_style_guide/general_format.html>

**Question One**

What are the main differences between a ‘data warehouse’ and a typical SQL database?

**Give examples and references.**

The main difference between data warehouses and SQL databases is in their purpose and structure. SQL databases are designed for operational tasks like recording daily transactions and optimized for many concurrent users. They store mostly current data in a structured format for quick retrieval. In contrast, data warehouses focus on data analysis, aggregating historical data from various sources for complex queries and insights. They prioritize read operations over writes, often denormalize data for faster analysis, and support fewer concurrent users due to the complexity of queries. A database can be looked at like a filing cabinet for daily records, and a data warehouse as a research library for historical analysis. (Donal Tobin, Mar 13, 2023, Data Warehouse vs. Database, [Link](https://www.integrate.io/blog/data-warehouse-vs-database-what-are-the-key-differences/#:~:text=Data%20warehouses%20are%20optimized%20for,operations%20ahead%20of%20write%20operations.))

**Question Two**

What are differences between someone tracking expenditures via an Excel spreadsheet versus an SQL database? (eg. scale)

**Give examples and references.**

Spreadsheets are simpler and suitable for small datasets, allowing basic calculations and visualizations. However, they struggle with scale, data integrity, and complex queries.

Databases like SQL excel with larger datasets, offering more robust data organization, security, and multi-user access. They enable advanced functionalities like relational connections, automated updates, and powerful search capabilities, giving deeper insights into spending patterns. If you were tracking 100 expenses in a spreadsheet. It's manageable. But for 10,000 expenses, a database is more suitable. (Editorial, July 5, 2023, Database vs. spreadsheet, [Link](https://blog.airtable.com/database-vs-spreadsheet/#:~:text=The%20main%20technical%20difference%20between,that%20come%20from%20external%20tables.))

**Question Three**

In compiled languages, what steps do programmers do to produce an executable file?

**Give an example.**

Compiled language programmers translate their code into machine instructions using a compiler. Then link these instructions with libraries to create an executable specific to the target platform. They can package the executable with resources for distribution. Python's PyInstaller tool shows this by creating standalone executables.

**Question Four**

What is the role of version control systems in software projects?

Give two examples of Github-like applications/ programs and cite their differences.

Version control systems act as the backbone of software development, meticulously tracking changes, enabling collaboration, and safeguarding code. They offer a complete history of modifications, allowing developers to revert to previous versions, analyze bugs, and understand design. Branching and merging empower simultaneous work and conflict resolution, while traceability connects changes to project management and issue tracking for informed decision-making.

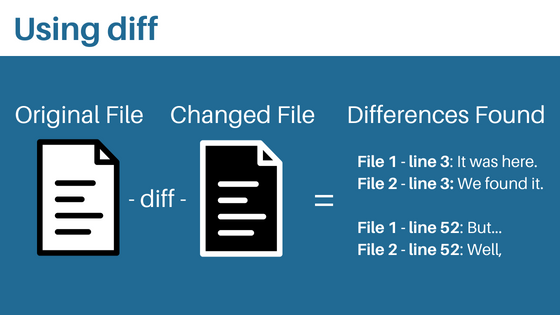
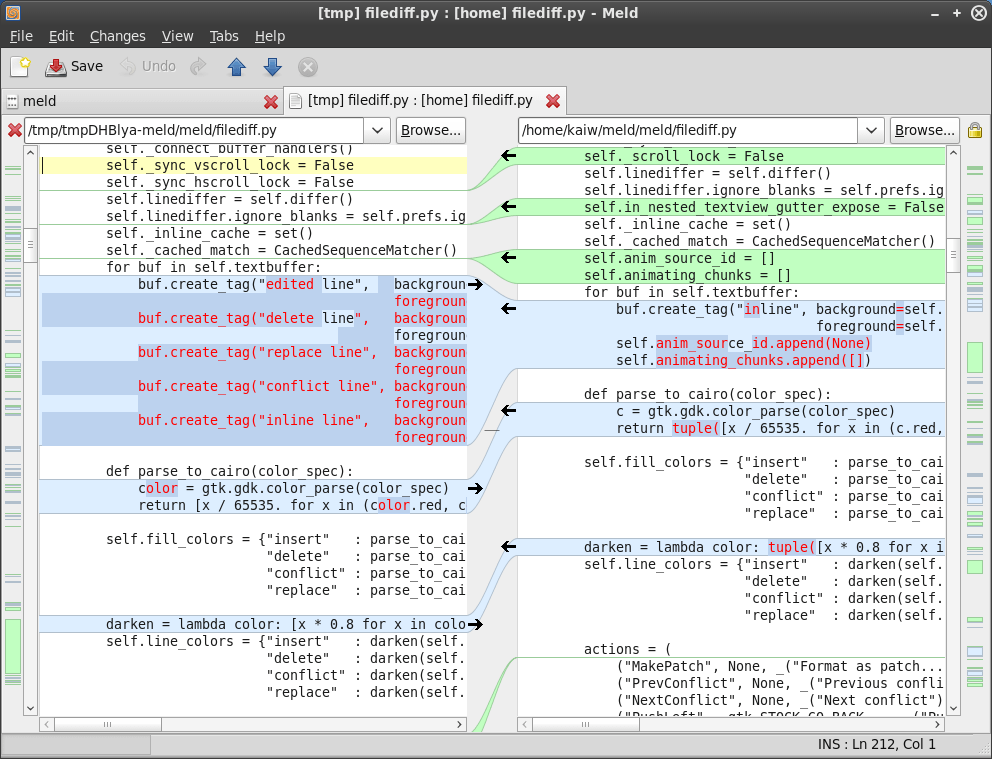
Two popular Git-like applications include GitHub which is a cloud-based platform primarily focused on public and private code repositories, emphasizing social coding features and community interaction. There is also GitLab, an open-source platform offering self-hosted or cloud-based options, catering to both public and private repositories. (Perforce, March 31, 2023, Overview of Version Control Software, [Link](https://www.perforce.com/blog/vcs/what-is-version-control)).

**Question Five**

What is a ‘**diff**’ file?

Paste a screenshot of a **diff** file here (showing side by side)

A "diff" file records the specific differences between two text files, typically used by software developers to update source code. It acts like a recipe explaining how to modify one version of the file to match the other. These files are often opened with programs like Mercurial, Kompare, or even text editors, but cannot be directly converted to other formats as they represent the changes themselves. (Tim Fisher, September 14, 2023, What Is a DIFF File?, [Link](https://www.lifewire.com/diff-file-2620661))

**Question Six**

Explain what a baseline is.

A baseline is a fixed reference point, like a before picture, against which you compare the new. In the IT world it typically includes three aspects: cost (budget), schedule (timeline), and scope (deliverables). By comparing current progress to this baseline, project managers can track performance, identify deviations, and improve future estimations. (Margaret Rose, 20 February, 2017, Baseline, [Link](https://www.techopedia.com/definition/6148/baseline))

**Question Seven**

How does the program version in the private workspace differ from the baseline version?

What is a conflict in terms of two different updates to a file? How does it get resolved?

Private workspaces in software development hold modified code for experimentation, separate from the stable baseline. They offer isolation to avoid conflicts when multiple people work on the same program but might be incomplete and lack documentation. Similarly, file update conflicts arise when separate changes affect the same section. Resolution involves manually reviewing or using tools to merge changes, choosing the best version, or discarding modifications depending on their importance and context. (Atlassian, Git merge conflicts, [Link](https://www.atlassian.com/git/tutorials/using-branches/merge-conflicts#:~:text=A%20conflict%20arises%20when%20two,to%20help%20resolve%20merge%20conflicts.))

**Question Eight**

What is the build and what is the result of the build?

A build, in software development, is the process of transforming human-readable source code into a runnable application, essentially taking the code and making it into a program. It can be done either from scratch which is a full build or by updating only changed parts which would be an incremental build. The main result of a build is a set of executable files binaries ready for testing, distribution, or even customer use. (Katie Terrell Hanna, March 2022, Build, [Link](https://www.techtarget.com/searchsoftwarequality/definition/build))

**Question Nine**

What is the three-tier architecture?

The three-tier architecture is a software design pattern that separates an application into three distinct layers: presentation (user interface), application (business logic), and data (storage). Each layer runs independently, allowing for faster development, improved scalability, reliability, and security. In web development, these layers translate to web server, application server, and database server. (IBM, February 14, 2024, What is three-tier architecture?, [Link](https://www.ibm.com/topics/three-tier-architecture#:~:text=Three%2Dtier%20architecture%20is%20a,associated%20with%20the%20application%20is))

**Question Ten**

What is polymorphism in technology? Give an example.

Polymorphism in technology gives programs the ability to treat different types of data and objects similarly, using the same interface but achieving unique results based on each object's specific characteristics. This allows for flexible, reusable, and adaptable code, simplifying development and improving the efficiency of handling diverse information. From user interfaces to complex simulations, polymorphism plays a crucial role in various software applications.

One function, "speak", makes a cat "meow", a dog "bark", and a bird "chirp" - using polymorphism to adapt to each animal's unique sound. (Sumo Logic, Polymorphism - definition & overview, [Link](https://www.sumologic.com/glossary/polymorphism/#:~:text=Ad%2Dhoc%20polymorphism%20refers%20to,the%20plus%20sign%20(%2B)%20operator.))

NOTE:

