**Wayne State University**

**Upload Date: Jan 24, 2023**

**CSC 4110 - Software Engineering**

**Weekly Homework Assignment**

**Directions:**

**Do all problems by the due date. Follow instructions explicitly. See submission requirements for each problem. This is an individual assignment. Don’t forget Questions at End of Assignment.**

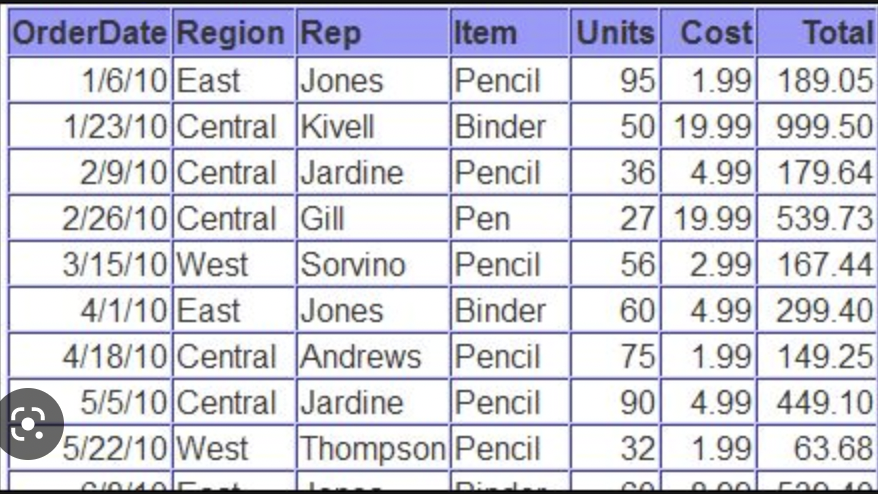
**Problem One**

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

**Assignment**

Write a program that **procedurally generates ‘sample’ data and stores that data in list form**, such as a ‘list of lists.’ Each entry will consist of order/vendor information such as usernames, web orders, product IDs, quantities, date of order, region, etc.. The specific column/ category names are up to you (see Requirements).

*See sample below.*



You will wind up employing various Pythonic strategies, tools, and methods in your syntax. Your methodology in how you accomplish this is absolutely 100 per cent up TO YOU.

For example, if you have a region column, you can have a ‘root list’ of the various regions (North, NorthWest, SouthEast, East) and employ the **random.choice()** method.

**DO NOT BE CONCERNED WITH FORMATTING DETAILS LIKE COLUMN MARKERS OR THINGS BEING CENTERED. This isn’t meant to look like a Microsoft Spreadsheet.**

**The goal is this:**

**If this code runs, does it PROCEDURALLY PRODUCE Sample Data, for the purposes of calculations and analysis?**

**Problem One Requirements:**

At least 5 columns of procedurally created data, stored as a lists, within an overall list.

The columns can be of any data, but should be made to look like a customer order list (date, product-ID, region, etc….)

There should be no less than 20 ‘rows’ of data auto-generated.

Details are left to the student; use past lectures and materials from texts provided.

**Problem One Submission Requirements:**

Copy/ paste the code image; copy/paste the output in this document. A screenshot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

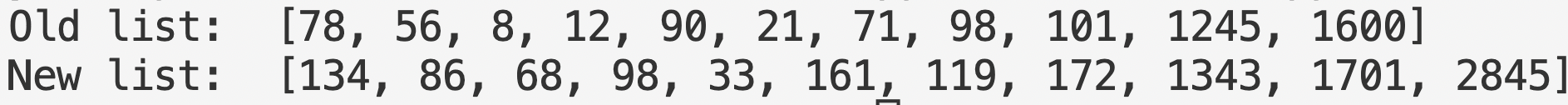
I just realized I put the labels for the wrong class as the header, to be fair I’m also currently in CSC 3110. Double checking I consistently made this mistake across all of the screenshots submitted, neat. 3 is just such a compelling number.

**Problem Two**

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

**Assignment**

Write a program that **uses automation to procedurally use adjacencies** to take value from the old list and create values in the new list. Recall that each neighbor in the new list is the sum of the immediate adjacent neighbors.



**Problem Two Submission Requirements:**

Copy/ paste the code image; copy/paste the output in this document.

A screenshot of a computer

Description automatically generated

I really need to stop naming my variables as the same name as the types, it leads to me making a lot of mistakes.

**Problem Three**

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

**Assignment**

Write a program that uses automation to procedurally separate five strings having a name and phone number. The result is two separate collections of names and phone numbers (they could be lists or strings).

Example:

START: user\_data="John Q Public 541-233-7612"

FINISH: name = “John Q Public” number=”541-233-7612”

**ProblemThree Requirements:**

At least 5 transitions to be demonstrated.

**Problem Three Submission Requirements:**

Copy/ paste the code image; copy/paste the output in this document.  
  
A screenshot of a computer

Description automatically generated

Wow it works!

**Problem Four**

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

**Assignment**

1. Print a string that uses double quotation marks inside the string.

2. Print a string that uses an apostrophe inside the string.

3. Print a string that spans multiple lines with whitespace preserved.

4. Print a string that is coded on multiple lines but gets printed on a

single line.

**Problem Four Submission Requirements:**

Copy/ paste the code image; copy/paste the output in this document.

A screenshot of a computer

Description automatically generated

**General Requirements:**

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

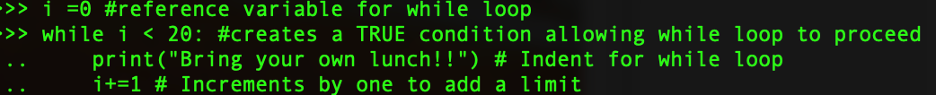
**AT LEAST ONE PROBLEM MUST USE SONIFICATION AND VISUALIZATION.**

#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**

**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.

**Discussion Questions**

**A FULL paragraph per question (between 4 and 6 meaningful sentence)**

**APA references ARE a MUST**

**Question One:**

What important properties does the initial development establish? Why are they important?  
  
The initial development decides properties such as the requirements, design, implementation, technology, and architecture of the planned development. The requirements specify what goals the project must actually accomplish, the design sets out how those goals will be generally accomplished, and the implementation is how the design will actually be put to code. Decisions such as technology choose the programming language and toolset (libraries) that the code will actually use. Architecture determines the style, components, and interactions that the design will actually possess.

**Question Two:**

How can code decay if it isn’t physical?  
  
Code decay is a misnomer, it is not that the code itself is decaying within the machine, it is that the functionality of the code within the organization it is used for is decaying. This decay can take place for multiple reasons – such as ever changing requirements turning what it initially simple code into a spaghetti monstrosity. Similarly, code decay can occur due to a loss of institutional knowledge within an organization – this can occur due to those experienced with the code retiring, being replaced, or moving onto other tasks within an organization. When these experienced members leave they take with them their understanding of the code they worked with, meaning that those attempting to work with the code now have a lesser understanding, and as a result the ability to work with the code decays, and the code’s functionality often decays as well.

**Question Three:**

Name three INTERNAL (inside the entity) factors affecting initialization and growth phases of such companies/ application as Rover.com and ADP, a cloud payroll service?  
  
1.) Shifting business priorities – Development and initialization of software may find itself cut short by the shifting priorities of the business developing it. For instance, SEGA canceled its planned SEGA 32X all in one console ‘Neptune’ due to the planned upcoming release of the SEGA Saturn. Similarly, Atari ended development on its 32 bit ‘Panther’ video game console in favor of development of the Atari Jaguar.   
  
2.)Stakeholder Needs and Requests – Software is developed for a purpose – and that is typically to satisfy the needs of the intended user. However, Stakeholder needs often change and thus the software must change with it.  
  
3.)

**Question Four:**

Name three EXTERNAL (outside the entity) factors affecting initialization and growth phases of such companies/ application as Rover.com and ADP, a cloud payroll service?  
  
External factors to a corporation affect the development and growth of companies and applications.  
  
1.) Legal/Regulatory climate: Development of certain products might find themselves facing additional hoops that make performing their intended task more difficult. This could include sudden changes in law permitting states to tax online sales even when a corporation lacks physical presence in the state, environmental regulators unexpectedly making the desired chemical product illegal, or changing accounting standards and tax codes necessitating an overhaul of a payroll service.  
  
2.)Business climate – Corporations often rely on credit to finance their continued expansion and development. It is not unusual for corporations to take out short term loans to finance projects. Changing fiscal conditions such as changes in the interest rate can therefore dramatically affect a corporations ability to procure capital – and as a result development of future products may end up being shut down.

3.) Changing technology conditions – A product in development may find itself obsoleted by changes in technology to the market. A product attempting to grow its market share may find itself cut out from the market by a better product. A strong example of this

**Question Five:**

What are the advantages and disadvantages of the prototype development model?  
  
The advantage of a prototype model is that it produces a working prototype before further expenditures are invested into a full version of the desired software. Having a prototype can demonstrate the viability of a product, and whether further investment is warranted. The completion of a prototype also allows for requirements to be corrected and for a better implementation to be implemented later down the line. However, the prototype model can be thought of as a versioned staged model with only two stages – prototype and development – this gives it weaknesses compared to the versioned-staged model because there isn’t continual new development to help account for actual use by the end-users instead of theoretical requirements. Similarly, the prototype model skips from implementation to maintenance, instead of continuously churning out new models – which means ultimately the prototype model will lead to faster obsolescence of the product, instead of prolonging the lifespan of a product that is finding success.