Fall 2023 ME 021: Engineering Computing

Instructor: Ayush Pandey

Python project ideas for ME 021 midterm exam

The following list consists of all student proposed and instructor suggested project ideas for the midterm:

1. Weather analysis - Be able to output Highest and lowest temperature recorded. Average temperature. Number of days it rained. Day's it said it wouldn't rain but there was rainfall. Visualize and report the data. Use real world data for the weather.
2. Real estate analysis - Make a histogram of the number of for sale houses in each state. Be able to output the most expensive house overall and also the most expensive house in a state chosen by user input and also average price in each state -
3. Movie rating analysis - Find the average movie rating for 3 movies based on their movie ID. There are 2 csv so will need to find the movie ID for the movie and then check the rating based on the movie ID in the other csv. Visualize the results.-
4. Employment analysis - Output Top 3 most employed_all_year majors. Output the average unemployed percentage as well as the average employed for a specific job. Be able to also output the average unemployed percentage as well as the average employed based on a major category -
5. Cook of a steak analysis - What is the average preferred cook on a steak percentage of those who voted. Be able to let users record their preferred choice as well -
6. Contact Book - Create a program that reads contacts from a file, allows adding, deleting, and modifying contacts, and then saves the updates back to the file.
7. Gradebook - Teachers can input student grades for various subjects, calculate averages, and retrieve student performance. -
8. Password Manager - A simple tool that allows users to store and retrieve passwords, ensuring they're saved securely (apply encryption).
9. Calculator - Create a calculator that has adding, subtracting, multiplication, and division. Store the latest three final answer calculations history in file and be able to output to let user see them if wanted -
10. Flashcard quizzer - Create a file with questions and answers. Have the user to be able to add a question and answer or to take a flashcard quiz and output a score

11. Rock Paper Scissors - Create a rock, paper scissor game and have a computer vs user score stored in a file. Before the game starts display the score and update the score according to who wins
12. Birthdays Reminder - Users can input their friends' birthdays, and store it in a file. The file should be ordered from recent to latest. Have the program be able to output the most upcoming birthday, the latest upcoming birthday, as well as specific the birthday of a specific person. All these outputs should also mention how many days away the birthday is from the current date.
13. Work logger - Create a file that stores and updates the amount of hours a person has worked in a day. The program should be able to output how much that person has made in a specified month by a user input by multiplying their hours by the minimum wage. Add visualizations, file reports, options for different employees, a live timesheet tool where you can continue adding your tasks. -
14. Dictionary creator - Create a dictionary where users can store words. Have user be able to look for specific words and also be able display all the words that start with a specific letter based on user input
15. Hangman - Create a hangman game that chooses a random word read from a file -
16. Tic Tac Toe - Create a tic tac toe game. Have a computer vs user score stored in a file. Before the game starts display the score and update the score according to who wins
17. Music Playlist Creator - Users can create and save playlists. Playlists can be loaded, edited, and saved back to files. Users may add statistics about the songs they listened to. Visualize the data of user listening history.
18. Address Book - Store and retrieve personal contact details. Users can add, edit, search, or delete contact details.
19. Student Grade Management System - The "Student Grade Management System" is a simple Python command-line program that allows users to manage and analyze student grades. This program provides various features to add, view, calculate averages, and save/load student grades to/from a JSON file. It's a useful tool for educators, administrators, or anyone who needs to keep track of and analyze student performance.
20. Movie Watchlist - Users can create a list of movies they want to watch, mark movies as watched, and maybe even rate or review them.
21. Number guesser - Randomly choose a number and have a series of prompts to say if they are very cold, cold, hot, very hot based on how far they are from the number. Store a history of how many tries it took for users for each game in a file. Before the game starts, display how many tries it took for the most recently played game -
22. Blackjack aka 21 card game - Implement a blackjack game and in the beginning of each game, display the score of computer vs user by reading from a file. When game is complete update score in file
23. Coin toss game -Create a coin toss game that displays the score and updates text file.

24. Recipe suggester finder - Given some ingredients that the user likes, suggest recipes. Create a menu of options that the user can explore and preferably create recipes that are realistic using a database -
25. News Aggregator - Using real-world news APIs or data, build a news aggregator that compiles top news articles based on user's preferences. Create a menu for the user and allow for easy navigation as the user reads the news. You may also build a dynamic scoring system for the news that the user likes (and is stored as files) so that if the same user uses your news aggregator again, they get more relevant news articles.
26. Currency Converter - Using publicly available exchange rate API or any other free currency conversion tool, create a currency converter that uses rates and converted amounts for selected currencies. Visualize the trends for a user of how the rates have changed in the past. -
27. Trivia quiz game - Using a database of trivia questions, the user plays a game and the score is then displayed. The user may be presented with a list of choices of the kind of trivia game that they would like to play. A trivia fact sheet can be written at the end of the game for the user on the various facts that the user may be interested in based on the questions they answered. -
28. Fitness tracker - Record the user's activity and diet for a day and show progress graphs, workout summaries, and dietary recommendations. Visualizations, record-keeping for multiple days, use of data on nutrition are all options that may be explored.
29. A simple e commerce platform - Display a list of products, user reviews, and user orders. Implement a online shopping cart functionalities, and order summaries to let the user "place" orders on your system. The orders can be recorded in files and/or visualized back to the user.
30. Russian Roulette - User plays a computer in a game of Russian Roulette.
31. Car sale finder - Find a users preferred car using a database/csv. Finds a user a car based on costs, ratings, etc. and recommends to the user based on what criteria they care about the most.
32. Password manager - Options to generate a password for a user, retrieve a password, manage multiple passwords on different websites, write passwords to a secure file and load them from files. Further exploration visualize the most commonly used passwords using data from the internet.
33. Adventure based text game - Displays a story with items that allow you to solve puzzles.
34. Baseball player analysis - Retrieves specific stats from a database of baseball players. Use real world data on player records and visualize the data. Crunch the numbers to spit out interesting facts.
35. A simple version of Steam, the gaming platform (SimSteam) - Users load in a file with the games that they own. The project lets the user make choices about those games so that details such as hours played, levels completed, and achievements/badges earned can

be recorded for each game. SimSteam would then report, visualize, and write outputs of consolidated information to a file for the user.

36. Task Scheduler - A scheduler that adds, edits, and deletes a task. Also displays a priority based on how far the task is away from the current data.
37. Recipe Cookbook - The project asks a user for recipe ingredients given a food name dish(Such as burger, chicken, enchiladas etc.). A constructed recipe is shown to the user received from a csv/txt file.. Create a menu option that allows users to either see all the recipes, find a specific recipe or add a recipe.
38. Workout Generator - Has a database/csv of workouts for different body parts. Lets users decide which body part to workout and recommends them potential workouts to do for them. Also can generate full workouts such as full body, upper body, lower body.
39. Car performance finder - Finds a user a car based on performance and recommends to the user based on what performance they care about the most.
40. Mortgage analyzer - Visualize the trends for the price of mortgage changes throughout the years. Lets the user enter the year, and how much they pay for mortgage to analyze potential price change in the following years. Will read from a file the set rate of charges such as expenses, utility, etc..
41. Activity Guide - Create a program that asks the user questions. These questions will be used to recommend users for activities to do in a city of their choosing. Will be reading these activities from a csv -
42. Air Quality checker - Display the quality of the air in your area. Will read for csv file to display previous checks done in that area before and will update file with current.
43. Electric vehicle charging cost calculator - The program will calculate the cost of changing an electric vehicle at different charging stations. It will take into account the battery capacity of the car, the desired battery percentage, and the distance they will travel -
44. Morse Code translator - Create a morse code translator program that saves the history of the words that have been translated in this program to a file. Have an option for the user to either see the list of words translated or to translate a new word.
45. Notepad - Create a notepad where the user reads and writes to a text or csv file.. Be able to append, find words, replace words, delete words.
46. QR Generator - Have a csv file with a person's name and their favorite links. Let users be able to add a person and their links. Let users choose a person and a specific link by reading from the csv, create a qr code and have it displayed on matplotlib. -
47. Palindrome checker - Be able to read from a csv file and go word for word and check if any of the words are Palindrome. Display the palindrome words found. -
48. Goldbach's conjecture - Program that lets user enter a prime numbers greater than 5 and gives out a combinations of two other prime numbers. Save history of imputed number and result to a text or csv file.
49. Song recommender - Ask the user question and recommend 3 songs that would be appropriate based on their choice. Will be reading from a csv/database that has a selection

of songs. Also have the option of adding songs straight from the script and prevent duplicates.

50. Make your own dice for board games - Built in dice of various kinds where user can choose and use a particular dice, with options for using it multiple times. The user could also create their own dice using this project for a particular game. All the dice rolls are recorded as files and the pictures of dice are shown as visualizations alongside the probability distribution graphs for the dice.
51. CatCard interface - Implement a simple Python duplicate of the CatCard MobileID website with options for adding funds, checking balance, reporting lost card, access to resources, and emergencies. Create categorized expense distribution charts to show the user where they spent the most money.
52. Pavilion menu generator - The UC Merced Dining office generates menu that is served at Pavilion for breakfast, lunch, and dinner daily. Your project would automate this menu generation such that nutritional contents are balanced, there are enough fun food items, drinks, and desserts. Write the weekly menu table to a file. For each menu (breakfast, lunch, or dinner) you may need to design the menu differently. -
53. Gacha simulator - Simulate different games gacha rates.
54. Wordle game - Create a wordle game. Uses a csv to receive words for the wordle game.
55. Speedometer Calculator - Has a database,csv of cities and their distances. Asks users their location and destination and time they want to spend to get there. Then calculates the speedometer needed to arrive at that time.
56. Video game recommender - The user will ask the user a series of questions, where they would search through a csv or txt file database to show the best video game recommendation for the user based on their inputs.
57. Battleship game - Create two random lists with ships(1's) and empty water (0's) randomly created in which the user and computer will try to hit the targets. Use a random module for computer gameplay. Before game starts, display the current score between computer and user
58. Bobcat Shuttle Tracker - Design a Python program that simulates the Bobcat Shuttle service, showing the current location options of shuttles. You can use loops to update shuttle positions and read from a text file for initial positions. Output could be a visual display of the shuttles and their next stops, possibly saved to a file.
59. Library Resource Locator - Implement a program that helps students find available resources in the Kolligian Library, such as study rooms, computers, or books. Use dictionaries to store resource information and availability. Outputs could include a map showing available resources and a feature to reserve them, with details saved to a file.
60. UCM Event Calendar - Create a Python program that fetches and displays upcoming events at UC Merced. You can load event data from a calendar .ics file and use functions to sort and display them based on categories like academics, sports, or social. Output would be an organized calendar, saved as a file for offline use.

61. Course Schedule Optimizer - Design a Python program that helps students optimize their course schedules based on preferences such as free days, preferred professors, or major requirements. Use lists to represent available courses and loops to find optimal combinations. Output would be a list of course combinations that meet the student's criteria, saved to a file.
62. Housing Roommate Matcher - Implement a roommate matching program for students living in university housing. Use lists to store student profiles and preferences and functions to find compatible matches. Output could include a list of potential roommates, which could be saved to a file for future reference.
63. Text adventure - Recognize player inputs and use them to maneuver the player through a virtual environment. Outputs will change according to player inputs and recognize invalid inputs. Puzzles requiring certain inputs in order to be solved (combination code) or that require the player input something before being able to interact. Timers that count down until player death with for loops. File manipulation to draw random responses.
64. Financial Aid Calculator - Create a program to help students estimate their financial aid and college costs. You would use conditional statements to account for different types of aid and scholarships. Output would be a detailed breakdown of estimated costs and aid, saved to a text file. -
65. ProfRateUCM - Design a simple review system where students can rate and comment on professors and courses. Use dictionaries to store review data and functions to calculate average ratings. Output could include a list of top-rated professors and courses, saved to a text file and appropriate visualizations.
66. Merced Outdoors Trip Planner - Implement a program that helps students plan outdoor trips around Merced. Use lists to store information about nearby hiking trails, lakes, and camping spots. Outputs could include personalized trip itinerary recommendations, saved to a file.
67. Alphabet game - Alphabet letter guesser, where a random letter would be chosen, and you would have to guess the letter based on whether it comes before or after the specific letter. Will have a timer that makes sure users only have a certain amount of time to give an answer or else game over. Will also log the faster time achieved into a text or csv File.
68. UCM Club Finder - Build a program that helps students find clubs and organizations that match their interests. Use dictionaries to store club information and functions to match student interests with clubs. Output would be a list of recommended clubs, saved to a text file for future reference.
69. Research Opportunity Finder - Build a program that compiles and organizes research opportunities for students. Use lists to store research positions and filters to sort them by department or required skills. Output would be a list of opportunities saved to a file. -
70. Cypher Decoder - encodes a message and produces a key, decodes a message with a key shifts different letters by different amounts a->c b->f c->z shifts each place of letter by a

different amount- the shift key for z might be first z->r second z->q so the word jazz might look like "JARQ"

71. Car building program - gives the user lists of cars and parts to choose from and will total the cost of everything in the end.
72. Akinator implementation - Create an akinator game where the computer guesses characters from a set list that the user is thinking of, including a log history of correct and incorrect guesses.
73. Music tracker - The user will be asked to input the song they want to play. Overall there will be a list of users where each user will have a history of played music. Each history will display artist name, song name, and rating, be able to have a ranking of songs from highest to lowest based on ratings and create a histogram using matplotlib displaying the rankings of the artist based on their mean of all their song ratings.
74. Poker probability - determine outcomes of winning poker hands from data set file and at the end determine number of wins each player had.
75. Gas trip requirement calculator - The project consists of asking the user for their type of engine and the liters of the engine.Will also have the option of letting users just input the name of the car as well. The user will then provide the number of miles that their trip is going to require and will output the amount of gas needed for that trip.
76. Wheel of Fortune - Emulate a wheel of fortune game where the prompt and the answer is received from a csv or txt file. Display final score in the end and store the latest game results in the file
77. DeskFinder - The user will be able to reserve a desk for 3 hours. It will show available desks to reserve, what desks are taken and for how much longer and also allow for future reservations. Desk reservation data will be received from a csv or txt file and updated to it.
78. Jeopardy game - create a jeopardy game that incorporates reading from a csv or text file.
79. English to Spanish Translation - Create a program that translates english words/sentences to spanish. Save the words to the document. Also, read from the file initially so that you look if the word or sentence has been translated before.
80. star catalog - the user would be prompted to choose to learn about constellations, star sizes, color, distance, etc.
81. Buzzfeed Quizzes - 3 mini quizzes (car quiz, computer quiz, and a phone quiz) plan on having it be multiple-choice. It will display A-D options which the user will type in A, B, C, or D and the code will tell you what you answered and will tell you if you are correct or incorrect. if you do not answer one of the questions it will have you restart the quiz from the beginning, have .txt files with all of the questions and answers each quiz will be a different .txt file.
82. Turbos and Superchargers - Write a program that finds the ways each charger works for a car, and how to improve each one by the code.

83. Geo Distance guesser - Will have 2 options, game mode and study mode. For game mode, it will have a list of locations in a csv or txt file and the computer will choose 2 random ones. The user will then have to guess the distance between them where they have to be within 20% of the correct distance value. For study mode you just let the user type in their own locations they want to see the distance of. For using maps and calculating distance you can use ‘from geopy.geocoders import Nominatim’ and ‘from geopy.distance import geodesic’
84. weightlifting exercises recommended - Recommend exercises based on user input. Will read from a txt or csv file that has the available exercises relating to specific muscle groups.
85. Fruit tree success calculator - Ask the user to choose a city and fruit tree from the provided list to calculate if it would thrive there depending on the USDA's hardiness zones, average rainfall, and average temperature. Also include if a plant should be regularly watered if there isn't enough rainfall. Use two text files, one including all of the city's information and one including all of the plant's information.
86. Soccer statistics app - Options for soccer players statistics, reports, visualizations and fun facts.
87. Python linter -
88. Largest Prime Factor - Computer the prime factor of a user's given number < 100. First check if the user's given number has been done before and so output their result, if not compute and output all the prime factors. Be able to store the given number as well as its largest prime number in a csv file.
89. Estimate Pi - The goal is to use random sampling to estimate the value of Pi. You'll use the Monte Carlo method to simulate points inside a square and a circle and then calculate the ratio to estimate Pi. Outputs would include the estimated Pi value and a plot visualizing the random points.
90. Fibonacci Sequence Generator - Build a program that generates the Fibonacci sequence up to a given number. You'll explore different methods for generating the sequence and measure their performance. Outputs would include the sequence itself and computation times for different methods. -
91. Newton's Method for Root Finding - Implement Newton's method to find roots of equations. You'll explore how initial guesses and tolerances affect the speed and accuracy of finding roots. The output would be the root values and a plot of the function.
92. Traveling Salesman Problem Solver - Write a program to solve the Traveling Salesman Problem using heuristic methods like the nearest neighbor algorithm. You could explore improvements or alternative algorithms. Outputs would include the shortest path and distance. -
93. Random Walk Simulator - Create a Python program to simulate and visualize a random walk in 2D or 3D space. You can explore how step size and dimensions affect the properties of the walk. Outputs would include plots or animations showing the walk.

94. Simple Encryption Algorithms - Build a program to encrypt and decrypt text using simple encryption algorithms like Caesar or Vigenère ciphers. You could explore the strengths and weaknesses of different encryption methods. Outputs would include the encrypted and decrypted messages.
95. Maze Solver - Implement an algorithm to solve mazes. You could use algorithms like depth-first search or breadth-first search and explore how maze complexity affects solving time. The output would be the solved maze, possibly visualized with the path highlighted. -
96. Choice game - create a game allowing making a bunch of choices that gives the user the freedom to choose which choice they decide on, and of course each choice leads to a different ending of the game.
97. Game Theory Simulations - Develop a program to simulate simple game theory scenarios like the Prisoner's Dilemma. You can explore how different strategies perform in repeated games. Outputs would include statistics like win rates for each strategy.
98. Simple Weather Simulator - Create a program that uses basic mathematical models to simulate and predict weather patterns like temperature and precipitation. You can use historical weather data as a base. Outputs would include predicted weather patterns for future days.
99. Loan Calculator - Develop a Python program to calculate loan payments and amortization schedules. You'll explore how changing interest rates and payment frequencies affect the total payment and term. Outputs would include the amortization schedule and summary statistics.
100. Statistics Calculator - Able to calculate statistical functions such as Mean, Median, Mode, Standard deviation etc.. given user input numbers. Choose a particular data set, visualize the results, and write the results to a csv file.
101. Enigma Machine - Replicate the functions of an enigma machine. Inputs to the program would be a string of alphabetic characters, which would output a second string of "scrambled" alphabetic characters, which could then be"unscrambled" by running it through the machine again. Would need to be able to find the combinations found on enigma rotors in order to accurately make it compatible with other enigmas.
102. Jumper.io - an app where security guards or like jumpers in events could add/remove people from a line and also see who is in the line as well as checking to see who has already paid to get in or not.
103. Invisible map adventure - Similar to Oregon trail the user will navigate through a map that they must keep track of themselves while avoiding traps. Will keep a death counter as well as a time to complete a tracker and displays it whenever a new game starts. Will be read from a csv.
104. Loot collector game - user will be controlling(with arrow keys; up, right, down, left) a spaceship(turtle) to collect loot. Each loot you collect with the spaceship will increase your score by 1. It will display your current score and your highest score.

105. Ping Pong game- Create a ping pong game that saves and receives scores to display during the beginning of each game.
106. color calculator - prime color calculator, where the user inputs their two desired prime colors, (Red, Yellow, Blue), then the program gives them what the two colors make. Can also add the secondary colors which add way more options to what can be made.
107. Timer reminder - a reminder that when you input a message and time it will show up the message at the time you input.
108. Fish Bait Sorter - Fish Bait Sorter provides the user with a premade list of Fish baits to choose from(25 total).The user can view the "prelist" if wanted as well. The user also has a "Favourites list", where the user can primarily add fish baits too, and if wanted . Another feature is for the user to check the type of bait, which it can be live or artificial. Many other mini functions are also present, like the fact that the user can't add the same bait twice, check their favorites list if it is empty, and etc.
109. Fast Food Ordering Simulator - User chooses a fast food restaurant of their choice. Then searches through csv and displays their menu. The user chooses what they want to order based on the menu and displays their total when finished.
110. Housing Campus converter - Program that calculates the expenses of food / living if you live on or off campus, also include the commuting cost as well taking inputs such as how many miles you live away from the school and current gas prices.
111. Average technological output of college freshman - An analysis that shows average time spent between academic and personal usage of different technologies. Will use matplotlib to visualize.
112. Age/Food Simulator - Creates a food ordering game where you input your age, height, gender, nationality, and weight, and based on this information, the age will help us determine the type of healthy or unhealthy food you like since younger people are more likely to eat unhealthy food than older people. So based on the ages of kids 5–12, teenagers 13–17, young adults 18–21, adults 18–65, and seniors 65+, we will ask for height and gender, and based on that, get statistics on how heavy people are usually in this age, gender, or weight group. Based on all that, get their nationality and weight, and based on the weight, figure out if they eat greasy or unhealthy food or if they don’t eat enough calories. Finally, at the end, do a survey asking questions on whether they feel like they are in a good position for their health.
113. Cake ordering form - Create a program that reads saves contact information.It will also calculate the price of the cake they ordered. It will also suggest pre made cake flavors.
114. Fun/Personality quizzes - Create a fun/personality test. There should be multiple personality tests in a csv where the user gets to choose which one they want to do such as which superpower is best for you, which Disney character are you, which The Office character are you etc.. Store the number of each result in the csv such as say 2 people got

the superpower of super strength, 4 people got laser vision, and 3 people got the superpower of flight, if someone takes the test and gets flight again then update it in the csv.

115. Class progress visualizer - Create a csv with a quiz of 5 or more quizzes with student grades. Have the user look up a quiz and output a histogram of the distribution of test scores from the class. (Use Matplotlib)
116. Product analyzer - Create a csv that has product name, number of sales, years. Let the user find a product and create a line plot to let the user see the history of purchases of the product to see if sales are declining or rising.
117. Attendance tracker - Read from a csv that has a class name and a list of 30 students and be able to create a row with a new date. Have it go through the list of students and let users be able to mark them down as present or not present. Also let users be able to add all the days each student has attended class and create a histogram graph to show their attendance rate. -
118. Create a pattern memorizing game - using the alphabet, have a pattern show for about 5 seconds, erase and then let the user guess what the pattern is. For each level, add 1 letter till the user enters an incorrect pattern. Upload game number, the total pattern, and the length of the total pattern to a csv file. Just keep the highest score, for every game read from the file and display the score to beat
119. Spell Checker - Let the user create a passage and have it run through a spell checker. List the words that need to be double checked for their spelling. Also be able to do the same by reading from a file and have it run through a spell checker. Go word for word from the list the spell checker gave and have the option of being able to change the word if the user says yes if not then skip. -
120. PR Weight Tracker - Lets user read from csv on current bench, deadlift, squat and displays/stores past pr’s and update if record is broken. Users add their weights through user input. Reports, visualizations, and logs are generated as needed.
121. scientific calculator - calculator that includes basic plus, mines, multiplication…also built in the calculator will be sin cos tan to calculate degree and allow the python file to import the calculation record to the txt file.
122. Dice game - 7/11 - Will create the 7-11 dice game and will use pygame to simulate the shuffling of the dice. The user will play with the computer and will keep a log of the score and have the game displayed.
123. Crime rate calculator - The program will allow the user to calculate the average number of crimes that occur per 100,000 people in a given area. The program will ask the user to input the total number of crimes that occur in an area and to input the total population of the area. The program will then give the user the average crime rate.
124. Stress Analysis on Materials - The goal is to simulate stress-strain curves for different materials. You would implement Hooke's Law and other stress-strain relationships in Python for conditions like tensile, compressive, and shear stresses. You could explore

how different materials react to stress, and you could extend this by introducing fatigue analysis. Outputs could include plotted stress-strain curves and a report generated in a text file. (Use Matplotlib for visual) -

125. Solar Panel Efficiency Calculator - The objective is to determine the efficiency of solar panels under varying environmental conditions. You could explore the impact of parameters like angle, weather conditions, and time of day. Outputs would include daily efficiency graphs and a summary report.
126. Electrical Circuit Solver - Simulate electrical circuits using Kirchhoff's laws. Explore how changing resistor values, adding capacitors, or inductors affects the circuit. Output could include voltage and current distribution across components, visualizations of the current and voltage with time, and reports saved as a CSV file.
127. Energy Consumption Reporter - Develop a Python program to model and optimize energy consumption in a building. You can explore the effects of insulation, HVAC settings, and external weather conditions. Outputs could include daily or monthly energy consumption charts.
128. Material Life Cycle Analysis - Develop a program to analyze the environmental impact of materials from production to disposal. You could explore different materials like plastics, metals, and composites. Outputs would include environmental impact scores and recommendations for material selection.
129. Message encoder - This program encrypts a message, and is able to send it to others where they are able to see it if by someone who has the key to decode it.
130. Skateboard part recommender - (Message student) Build a website that recommends the correct part to build your own skateboard.
131. Logo Guess quiz - The user will build a program that lets user decide which logo is the correct one
132. Text Adventure game - create a text-based adventure game where the user must collect 3 tokens to complete the game. Each token must be earned by completing a mini-game, one will be a word game, one will be a number game, and the third will be a puzzle.
133. Food recommender - This recommender will take the qualities you want in food and will print out some food that have the qualities.
134. Housing Roommate Finder -
135. Housing Roommate Finder -
136. Recommend Video Games based on user selection of interests - Finding out what type of platform/console the user likes to play on, what types of genre of video game the user likes to play, how many people the user like to play with, the duration of how long each game takes to complete
137. Periodic Table lookup - The user will enter the name of the element or symbol and will output all the information of that element.
138. Bobcat Shuttle Tracker -

139. Bobcat Shuttle Tracker -
140. Vehicle Service Tracker - You will enter the Mileage, Year, Make, and Model and store the vehicle service history of it. (Get more info from student) -
141. Go Fish card game - Implement a Go Fish card game where you will also keep a history of wins and losses vs a computer.
142. Soccer Code - (Email student for a better description and title) -
143. Russian Roulette -
144. Higher and Lower Game - There will be a number and the user will receive random numbers generated in between. The user will receive a random number and type either higher and lower until the user matches the number. Will keep a log of the total time it took to guess the number.
145. Blackjack - The code creates and manipulates a deck of cards for a simplified version of the card game blackjack. It defines card values, generates a deck of cards with varying values and suits, and computes the overall value of a player's hand, including specific rules for dealing with Ace cards. The code allows you to play blackjack and alter the value of Aces as needed to keep the total value of the hand within the game's limitations.
146. Trajectory Calculator - Create a calculator capable of calculating trajectories using an initial velocity vector in 2d and 3d space.
147. Shoe size comparer - Create a program that compares shoe size split into men and women.Compare data and find patterns.
148. Food Ordering simulator -
149. Club Finder - Use dictionaries to store club information and functions to match student interests with clubs. Output would be a list of recommended clubs, saved to a text file for future reference.
150. Energy consumer - Energy consumption reporter, it will analyze the amount of energy that is consumed in a building.2 Write down the project inputs and outputs will include user input for IVAC settings and external weather while outputs could be daily or monthly energy consumption.
151. Personal Finance summary - Determine all the expenses for a user and set a goal for how much they want to save for the month. The code will ask users for their expenses, such as cost of all the spending they do whether it be rent or on fun activities, it will need income for the month and it will need the amount of money they want to save. The program will then show the user what they need to cut out to save money.
152. Athlete Player stats - Create a self-rating program for athletes that play multiple sports. This will help players understand where they are as a player. The user will input an Athlete's honest approximation of their own skills related to their sport and the program would give the player an approximate overall rating of their skills.
153. Time Management App Project - Create a mobile app for time management.
154. Rock paper scissors game -

155. Rock Paper Scissors - Create a rock, paper scissor game and have a computer vs user score stored in a file. Before the game starts display the score and update the score according to who wins
156. GI Character Analyzer - Print detailed stats about Genshin teams. The program will need Characters, Weapons, Levels, Effects, Loads of Genshin stats.
157. Estimate Pi - The goal is to use random sampling to estimate the value of Pi. You'll use the Monte Carlo method to simulate points inside a square and a circle and then calculate the ratio to estimate Pi. Outputs would include the estimated Pi value and a plot visualizing the random points.
158. Interactive Program - Program dealing with directional Movement and QE for rotations.
159. Workout Planner - Goal topics for the planner are to take the number of days per week and split them up evenly. Give exercises for certain muscle groups to target, space them out with optimum recovery time, Provide exercises for the muscle groups being targeted. Find out if the person wants to cut/maintain/bulk on their diet. Give a list of the average calorie intake required for men and women.