AP COMPUTER SCIENCE PRINCIPLES PORTFOLIO

XAVI

ABOUT ME

My name is Xavier Cutlip-Mason, and I'm passionate about technology, athletics, and academic excellence. I play ice hockey for the Piedmont Predators AA team and Ultimate Frisbee for Yorktown's B team, where I love the competition and teamwork that drive both sports. Academically, I've maintained straight A's every quarter since sixth grade, earning the Principal's Award and the NVSHL Student-Athlete Award. Beyond the classroom, I immerse myself in robotics, programming, and 3D printing, constantly exploring new innovations and challenges. My ultimate goal is to pursue a career in robotics and environmental engineering, combining cutting-edge technology with sustainability to create solutions that make a real impact on the world. Whether on the ice, the frisbee field, or designing new creations, I'm always pushing myself to learn, improve, and innovate.

GOALS

1

Master foundational coding skills: Strengthen programming logic, syntax, and problem-solving.

2

Apply computer science to real-world issues: Use technology to tackle sustainability, environmental engineering, and automation.

3

Prepare for future learning: Build a strong base for advanced robotics, engineering, and software development.

MY EXPERIENCE

Throughout AP Computer Science Principles, I faced numerous challenges, from grasping complex coding concepts to debugging stubborn errors. At first, programming felt overwhelming, and there were moments of frustration when projects didn't work as expected. But instead of giving up, I leaned into problem-solving, sought help from classmates and teachers, and practiced consistently. Over time, my confidence grew, and coding became more intuitive. By the end of the course, not only had I developed a strong foundation in computer science, but I had also learned a valuable lesson in perseverance, understanding that setbacks are just stepping stones to mastery.

CODING EXAMPLES

Flashcard App

Encrypt/Decrypt Software

Password Generator

Helpline Chatbot

FLASHCARD APP

This Python flashcard app functions similarly to Quizlet, providing an interactive learning experience with customizable flashcards. Users can add, delete, and shuffle flashcards to enhance their studying process. It includes a reverse mode, allowing users to flip questions and answers for varied learning approaches. The app features a testing mode, where users are evaluated on a set of flashcards and given feedback on how many they got right. Additionally, the "Track Progress" feature identifies mastered flashcards and allows users to redo the ones they struggled with. To ensure continuity, the app supports saving flashcards, enabling users to access and modify their study sets later.

ENCRYPT/DECRYPT SOFTWARE

This Python program encrypts and decrypts text using the Caesar cipher, shifting letters by a specified amount. The “caesar_cipher” function processes alphabetic characters while preserving non-alphabetic ones. The main function gathers user input and determines whether to encrypt or decrypt by adjusting the shift value accordingly. Running in a loop, the program allows repeated encryption and decryption until the user chooses to exit.

PASSWORD GENERATOR

This program creates a customized yet randomized password based on user input, including their name, pet's name, birth year, and a special character. The input is shuffled to increase randomness, and additional security measures are applied—such as introducing uppercase letters and random digits if none exist. The password generation process leverages Python’s random module to ensure unpredictability. The program operates in a loop, allowing users to repeatedly generate new passwords until they choose to exit, making it both interactive and user-friendly.

HELPLINE CHATBOT

This program creates a compassionate chatbot that engages users in supportive conversations while analyzing their input for distress signals, tailoring responses based on age and gender, suggesting calming activities, and providing urgent crisis intervention when necessary. It employs Python's regex module and predefined keyword lists to detect suicidal ideation, anxiety, loneliness, and violent language, advising users to seek professional help or emergency services if needed, while ensuring safe interaction by ending conversations when harmful speech is detected. The program operates in a loop, allowing continuous dialogue until users exit voluntarily or the chatbot identifies critical distress signals, making it an interactive and empathetic experience.

USE OF LOOPS

```
while True:
```

This loop ensures the program repeatedly prompts the user for input, allowing them to generate multiple passwords until they choose to exit. It provides an interactive experience instead of running once and terminating.

```
while True:
```

This time the loop keeps the conversation going until the user decides to exit. It ensures continuous interaction rather than terminating after one response.

```
for char in text:
```

This loop iterates through each character in the input text, shifting letters while leaving non-alphabetical characters unchanged. It ensures that every valid character undergoes encryption or decryption.

USE OF FUNCTIONS

```
def generate_password(name, pet_name, birth_year, special_char):
```

This function centralizes password generation logic, keeping the main program organized and allowing easy reuse. It improves security by randomizing characters in a structured way.

```
def caesar_cipher(text, shift, mode):
```

This function encapsulates the encryption/decryption logic, improving readability and making the code easier to maintain or modify. It allows users to apply a shift transformation based on their chosen mode.

```
def analyze_user_message(user_message, age, gender):
```

This function processes the user's message, categorizing it based on emotional keywords to generate appropriate responses. It improves response personalization and chatbot effectiveness.

USE OF LISTS

```
result = ''
```

Although result is technically a string, it is built dynamically like a list, with characters appended sequentially before forming the final encrypted message.

```
"suicidal": {
    "keywords": [
        "suicidal", "kill myself", "hurt myself", "end my life", "can't go on", "no point in living",
        "die", "wish i wasn't here", "hopeless", "don't want to wake up"
    ],
```

This list stores predefined keywords for detecting concerning language in user input. It allows efficient checking and response generation based on emotional distress indicators.

```
shuffled_string = ''.join(random.sample(combined_string, len(combined_string)))
```

The random.sample() function treats the string as a sequence of characters, shuffling them like a list before reassembling them into a new password. This prevents predictable password structures.

USE OF IMPORTS

```
from tkinter import filedialog, messagebox
import pygame
import random
import sys
import time
import os
import colorsys
import json
```

These imports extend Python's functionality: "tkinter" for GUI interactions, "pygame" for game development, "random" for unpredictability, "sys" for system control, "time" for delays, "os" for file management, "colorsys" for color conversions, and "json" for structured data handling.

```
import random
import string
```

These imports provide access to essential functions for randomization and predefined character sets, making password generation more secure and diverse.

```
import re
```

The re module allows advanced pattern matching to detect harmful or sensitive phrases. This ensures proactive response handling and alerts when necessary.

USE OF ERROR HANDLING

```
except ValueError:
    print("Invalid input. Please enter an integer.")
```

This prevents errors when the user enters a non-integer shift value. Instead of crashing, the program provides guidance on correct input format, improving user experience.

```
except Exception:
    category = "adult"
```

This error handling ensures that if an unexpected issue occurs, the program safely assigns "adult" as the default category rather than crashing. While it prevents runtime errors, a more specific exception handling approach would improve debugging and accuracy.

```
if not any(char.isdigit() for char in password):
    password += str(random.randint(0, 9))
```

This ensures the password contains at least one digit, addressing a possible oversight where the randomization process might result in a password with only letters. It adds robustness to password security.

AHA

One of my biggest “aha” moments was realizing how algorithms shape everyday life, from social media feeds to search engine results. Understanding how small optimizations can drastically improve efficiency changed how I approach problem-solving.

CHALLENGE

Recursion was one of the hardest concepts to grasp at first, as thinking about functions calling themselves was counterintuitive. Breaking problems down visually, tracing function calls, and writing small recursive functions helped me build confidence in applying recursion effectively.

CHANGED UNDERSTANDING

Initially, I thought computer science was mainly about coding, but now I see it as a problem-solving discipline that applies logic and creativity to real-world challenges. I also gained a deeper appreciation for abstraction and how layers of technology interact.

FUTURE GOALS

As a future mechanical and environmental engineer, I plan to integrate computational methods into sustainable energy, efficient design, and environmental monitoring. Using simulations, automation, and data-driven solutions will allow me to innovate eco-friendly systems that balance functionality with sustainability.

DIGITAL CITIZENSHIP

Topics like data privacy and algorithmic bias made me more aware of ethical concerns in engineering simulations and automation. Learning about intellectual property highlighted the balance between innovation and protecting creators, which is crucial in designing new technologies. Thinking critically about how computational models influence real-world decisions, from climate policy to resource management, reinforced my responsibility to develop ethical, sustainable solutions.

THANK YOU

APCSP PORTFOLIO