Tanuska Biswakarma

4688 Red Lion Place, Lawrenceville, Georgia 30044
Cell: (470) 223-9724
Email: tanuskabiswakarma@gmail.com
LinkedIn: www.linkedin.com/in/tanuska-biswakarma-5a01a7346

INTRODUCTION AND CAREER OBJECTIVE

Currently I am pursuing bachelor's degree in computer science from Georgia State University, that I expect to finish next year. My plan is to do master's to enhance my skills in the fields of Database, Data analyst, AI, Cloud computing and coding which I am very passionate about. My objective is to learn and get into these fields, which in the coming days will have a big impact everywhere and I want to be part of it and would love to contribute my ideas.

I am an organized and dependable candidate, successful at managing multiple priorities with a positive mindset and attitude. Willingness to take on added responsibilities to meet goals. I am good with communication and teamwork with ability to quickly learn and adapt to new environments. I love to further develop my skills and am committed to continuous learning and growth.

SKILLS

- Basic CentOS v7, Windows v10, MacOS experience.
- Working with Virtual Machine (VM) with Virtual Box v7 application.
- Basic programming with MySQL v9.1 with DBeaver client software.
- Basic working with Python v3.1 and Scratch programming knowledge.
- Basic C programming knowledge.
- Basic programming with Java v17.
- Experience in Microsoft Word, Excel, Powerpoint, Outlook and Publisher.
- Experience with WeVideo, Adobe Photoshop.
- Experience with CANVA application.
- Experience with Autocad v2019.
- Ability to work independently or as part of a team.

WORKING EXPERIENCE

Assistant Librarian | Guilderland Public Library, Guilderland, NY

I got an opportunity to work at Guilderland Public Library as Assistant Librarian, where I learnt how to organize and catalog books, video games etc, checking them in and out using their catalogue application. I also helped people who approached for help on a day-to-day basis. I worked under Brayana Wachowicz, who was from Youth Librarian, Programming and Public Services.

COLLEGE PROJECTS

SQL Programming:

1. **Project Name:** Student Management System Database Setup

Project Aim: The aim of this project is to create and configure a relational database in MySQL for managing student information, departments, course enrollments, and courses. This database will facilitate student record management, course registrations, and departmental organization.

Project Challenges:

- I had to consider the Database Configuration; the hardest part was setting up MySQL and configuring dBeaver correctly and then ensuring the correct database is selected as default.
- I had to consider the SQL Script Execution, making sure that I wrote the SQL scripts correctly to create tables and then avoiding any syntax errors and ensuring correct data types.
- I had to consider the Data Integrity and Structure, making sure to define appropriate primary keys and ensuring constraints like NOT NULL are correctly applied.
- I had to consider validation and debugging, making sure my scripts ran without errors and ensuring all tables are correctly created and accessible.

2. **Project Name:** Employee Payroll Management System

Project Aim: The goal of this project is to set up and manage an Employee Payroll Management System database using MySQL and dBeaver. This includes creating tables, loading data, and integrating a Java program (GetEmployeesPayroll.java) to fetch employee records and payroll details dynamically. The project involves writing SQL queries and Java code to retrieve and process payroll data efficiently.

Project Challenges:

- I had to consider the Database setup in dBeaver, making sure to correctly set up employeeData as the default database and then running SQL scripts to create tables and insert employee records.
- I had to consider the SQL script execution and data management, ensuring correct execution of employeeData_MySQL_create.sql for table creation.

Running employeeData_INSERT_datum.sql to load initial data. And handling potential SQL errors like missing constraints or incorrect data types.

- I had to integrate Java with MySQL, understanding and modifying the GetEmployeesPayroll.java file. And implementing getPayroll() to retrieve payroll details dynamically.
- I had to consider code refactoring and testing, removing redundant code (getPayroll() method) from GetEmployeesPayroll.java. and debugging Java SQL queries to ensure correct payroll retrieval.

Python Programming:

1. **Project Name:** Guess My Number

Project Aim: Prompts the user for an integer, tells the user to think of a number between 0 and n - 1, then makes guesses as to what the number is. After each guess, the program must ask the user if the number is lower, higher, or correct.

Project Challenges:

- I had to consider if the user may enter an invalid or unexpected input (e.g. letters instead of numbers, or an empty string).
- I had to consider whether the user may misunderstand the instructions or provide inconsistent responses (e.g. always answering "too high" or "too low").
- I had to consider if the user gave inconsistent responses (e.g. alternating "too high" and "too low"), the loop might never terminate.
- I had to consider the program may not scale well for very large values of n due to increased input size or response time.

2. **Project Name:** DeDuplication

Project Aim: The program takes an input list [50, 11, 33, 21, 40, 50, 40, 40, 21] and generates an output list [11, 21, 33, 40, 50], which removes the duplicate elements and sorts the list.

Project Challenges:

- I had to consider making sure the program handles all types of valid and invalid inputs gracefully such as empty lists, lists with negative numbers, lists with mixed data types (e.g. integers, strings, floats).
- I had to consider avoiding excessive memory usage when the range of numbers in the input is large (e.g. a sparse list with a very high maximum value).
- I had to consider whether the program works correctly for edge cases like single-element lists, lists with all identical elements, negative numbers or zeros, large lists with repeated values.

- I had to consider if the program needs to deduplicate non-numeric data (e.g. strings or mixed data types).

3. **Project Name:** Cashier

Project Aim: Calculate the change you are due when you buy an item in a store.
Pre-conditions (input):

You give some money to the cashier. Then, the cashier returns you the change.
i.e. The cashier should return you $1.35.

Post-conditions (output):

Change user get back from the cashier (dollars, quarters, dimes, nickels and pennies)
i.e. How many dollars, quarters, dimes, nickels, or pennies makeup $1.35? $1.35 contains 1 dollar and 5 quarters

Project Challenges:

- I had to consider making sure the users enter valid data, such as numeric values for cost and payment and handling invalid inputs (e.g. letters, special characters) gracefully without crashing the program.
- I had to consider how the program will detect when the amount paid is less than the cost, while providing a clear message to the user about the insufficient payment.
- I had to consider adapting the program to handle different currency formats, such as symbols, decimal separators, or rounding rules in other regions.
- I had to consider that the program is tested with a wide range of scenarios, including edge cases and debugging unexpected behaviors that might arise from incorrect calculations or user input.

4. **Project Name:** Password Generator Project

Project Aim: To create a program that generates secure, random passwords. The passwords must have a mix of uppercase and lowercase letters, numbers, and special characters (symbols). It Allows users to specify the desired password length. Ensures passwords are strong and secure by including diverse character types. And Provides users the option to accept or regenerate passwords until satisfied.

Project Challenges:

- I had to consider the randomness of the generated password to make it unpredictable and avoid patterns or biases due to how the characters are selected.

- I had to consider dealing with scenarios such as extremely large password lengths (e.g. 1,000 characters) or a length of 0 or a non-positive value. And making sure the program behaves predictably in these cases.
- I had to consider avoiding predictable patterns in password generation and educating users on the importance of a strong, diverse password without overwhelming them with technical details.
- I had to consider writing clean, well-structured code that is easy to understand and modify, also avoiding unnecessary repetition, such as generating the password multiple times in different ways.

C Programming:

1. **Project Name:** My Little Pony

Project Aim: To create a C program to display song lyrics in a random order. Prompt the user to enter an integer number, then randomly display a line from a song, and repeat this according to the integer.

Project Challenges:

- I had to consider that the program does not validate user input. If the user enters a non-integer or a negative number, the program may behave unexpectedly.
- I had to consider the lyrics are hardcoded into the program. If I want to modify or expand the lyrics, I need to recompile the program.
- I had to consider that the user knows what input to provide. A better user experience might involve clearer prompts or handling invalid inputs gracefully.
- I had to consider that while this program doesn't allocate dynamic memory, it's important to ensure that all resources (e.g. file descriptors or dynamically allocated memory) are managed properly in larger programs.

2. **Project Name:** Cheese Sorting

Project Aim: C program that manages a list of cheese names, sorts them alphabetically, and prints the sorted and unsorted lists for different test cases. The program will include functions for printing a list of cheese names, sorting the list of cheese names in alphabetical order, testing the program with various initial orders of the cheese list, including ascending, descending, and random order.

Project Challenges:

- I had to consider that the program does not check for invalid inputs when interacting with the user. If the user inputs incorrect or malformed data (e.g.

entering a string when expecting an integer), it could cause the program to crash or behave unexpectedly.

- I had to consider the test cases (ascending, descending, and random order) that are hardcoded into the program. If the list of cheeses or the number of cheeses changes, the code must be manually updated.
- I had to consider if the list of cheeses contains duplicates or is already sorted, the program will output repeated lines or display the same result multiple times.
- I had to consider testing small, medium, and large lists of cheese names to help identify potential performance bugs. For example, how the program handles large input sizes or very short cheese names could be critical.

3. **Project Name:** Seed value with random values

Project Aim: C program that continuously prompts the user for a seed value and generates five random numbers using that seed. If the user enters 0, the program will exit the loop.

Project Challenges:

- I had to consider if the user generates a large number of random numbers or uses complex algorithms for randomness, the program's performance could degrade.
- I had to consider how the program terminates only when the user inputs 0. While this is a reasonable choice, it might confuse users who expect an explicit "exit" command or want a different way to stop the program.
- I had to consider that code written on one platform may not work the same way on another due to differences in compilers, operating systems, or hardware.
- I had to consider that as the program grew, maintaining readability and ensuring the code remains understandable and modular.

4. **Project Name:** Random Number Generator

Project Aim: C program that generates random numbers between 1 and 100, keeps track of their frequency, and then allows the user to inquire about the frequency of a specific number. It also calculates the percentage of odd numbers that appeared in the 50 random draws.

Project Challenges:

- I had to consider that accessing array elements outside their valid range can cause undefined behavior, memory corruption, or crashes.

- I had to consider that it's essential to make sure that the program behaves as expected under various conditions. Users might enter invalid values, such as out-of-range numbers or non-numeric input, leading to confusing or incorrect behavior.
- I had to consider that this program is calculating the percentage of odd numbers. Since percentages are floating-point numbers, precision issues may arise due to how floating-point arithmetic works in C.
- I had to consider that as the program grows, performance can become an issue. The current program works well for 50 numbers, but generating larger datasets or performing more complex calculations might slow down the program.

AWARDS

- President's Education Award
- Senator George Amedor Jr, Certificate of Achievement
- National Junior Honor Society
- Georgia State University President's List Fall 2022
- Georgia State University Dean's List Fall 2023
- Georgia State University Dean's List Spring 2023
- Georgia State University Dean's List Spring 2024
- Guilderland High School Board of Education Scholastic Award 2017 and 2018
- HOBY New York East Leadership Award
- SWAG Award
- Certificate of Award by New York State School Music Association

EDUCATION

Bachelor of Computer Science (Currently pursuing, completion in May 2026)

Georgia State University, Atlanta, GA

- Coursework:

  Principles Of Computer Science I, Principles Of Computer Science II, Theoretical Foundations Of Computer Science, Computer Organization And Programming, Design & Analysis: Algorithms, System Level Programming, Operating Systems or Programming Language Concepts, Software Development, Data Structures, Discrete Mathematics, Calculus Of One Variable I, Calculus Of One Variable II

Alpharetta High School, Alpharetta, GA- *Honors and AP Classes* **-** ***High School Diploma,*** **May 2022**

HOBBY AND ACTIVITY

- Reading novels, Listening music, JAVA Club, JustRun Club.