

Project ideas

Project 1: Pattern Printing Program with User Input

Description:

Create a program that generates various patterns using loops and characters (e.g., stars, numbers) based on user input. Users can specify the pattern type, size, and character to use.

DoD (Definition of Done):

Accept user input for pattern type (e.g., square, triangle), size, and character.
Implement pattern generation using loops based on user input.
Provide a user-friendly interface to display the generated pattern.

Project 2: Customer Data Sorting App

Description:

Develop an application that reads unordered customer data from a text file, sorts it based on Customer ID using Quick Sort and/or Merge Sort algorithms, and saves the sorted data in a separate file.

DoD (Definition of Done):

Create a file reader to fetch customer records from the input text file.
Implement Quick Sort and/or Merge Sort algorithms to efficiently sort customer data by Customer ID.
Create a file writer to save the sorted records into a new text file.

Project 3: Interactive Dictionary Application

Description:

Build an interactive dictionary application using a tree data structure. Users can search for words, get word definitions, and receive suggested words if the input word is not found.

DoD (Definition of Done):

Create an index page for words from A to Z, making word retrieval efficient.
Implement a search function to find word definitions based on user input.

Project 4: Comparison of 'random search' and binary search technique.

Description:

Create a list of 10,00,000 numbers using random number generation function (in Java a Random class is there to handle this).

Output: the time to search a given number (i) in case of random search (ii) in case of binary search

DODs:

- a. Apply random search. Generate an index (a random number between 0 to 9,99,999) and check the desired number availability at that location. If found, print the time otherwise repeat the process and search again.
- b. Sort the list using any in-built sort() method and apply binary search. Search the same number as in case of random search and print the time to search the number.
- c. Run the program to search for different user defined numbers.
- d. List the time for both kinds of search and the difference between them.

Project 5: Check the efficiency of singly linked list and circular linked list

DODs:

- a. Create a singly linked list of 5000 nodes. Data may be inserted using Java Random() class method
- b. Create a separate circular linked list of same nodes as taken for singly linked list
- c. Take 10 numbers which should exist in the list and search in the singly linked list. Every time the search should start from the beginning of the list.
- d. Search the same 10 numbers in the circular linked list but next number search should start from the node where the first search stops.
- e. Compare the time difference between them.

Project 6: Stack-Based Calculator

- **Description:** Create a calculator that uses a stack data structure to evaluate arithmetic expressions, including infix to postfix conversion.
- **Definition of Done:**
 - User can input arithmetic expressions with basic operators (+, -, *, /).
 - The calculator can convert infix expressions to postfix notation.
 - The calculator can evaluate postfix expressions.
 - The application handles invalid expressions and provides meaningful error messages.

Project 7: Student Gradebook

Description: Develop a gradebook system using data structures like arrays or dictionaries to store student data and grades.

Definition of Done:

User can add students with names and assign grades.

User can calculate and display class statistics (average, highest, lowest).

User can retrieve individual student grades.

The application handles missing data and provides informative reports

Project 8: Student Academic Performance Analysis

Description:

Create a system for analyzing student academic performance using data structures to track grades, attendance, and extracurricular activities.

Project 9: Route Optimization for Delivery Services

Description:

Develop a route optimization system for delivery services that employs data structures like graphs and algorithms like Dijkstra's or A* to find the most efficient delivery routes

Project 10: Rock-Paper-Scissors-Game

Rock paper scissors is a hand game usually played between two people, in which each player simultaneously forms one of three shapes with an outstretched hand. These shapes are "rock", "paper", and "scissors". This project is aimed at automating one of the players called the Computerized Player and taking rock, paper or scissors as an input from the user. Automate all rules of the game.

Project 11: Shortest metro route prediction

Create a menu driven program to assist passengers in finding the shortest route (Time and route information to be displayed) in Metro.

It should cover all metro stations and their distances. Consider all real time constraints for finding routes.

Project 12:University Exam Scheduling

Creating an exam scheduling project using graph data structures can be a challenging and rewarding endeavor. In this project, you'll design a system to efficiently schedule exams for multiple courses and students, taking into account various constraints and preferences.

Project 13:Time table Scheduling

In this project, you'll design a system to efficiently schedule exams for multiple courses and students, taking into account various constraints and preferences such as:

One subject can have only 1 class in a day.

tutorials can have 2 slots one : 1st and two : 6th if required

Lab takes up 3 slots (3 hours as same as institute)

Lectures can take 7th to 12th slots.