A Problem Solving and Programming Course Project Report

**“Student Record Management”**

*SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF*

**BACHELOR OF TECHNOLOG IN**

## “DEPARTMENT OF ENGINEERING SCIENCE AND HUMANITIES”

**OF**

**VISHWAKARMA INSTITUTE OF TECHNOLOGY**

## Savitribai Phule Pune University

**Isha Thakur (12311945)**

**Raj Thakur (12310176)**

**Shatananada Thakur (12310748)**

**Prashant Thorve (12311619)**

**Anshul Thote (12311942)**

**Tanishq Thuse (12310237)**

*UNDER THE GUIDANCE OF*

**“Prof. Lokesh Khedekar Sir”**



## DEPARTMENT OF ENGINEERING SCIENCE AND HUMANITIES

BANSILAL RAMNATH AGARWAL CHARITABLE TRUST’S

VISHWAKARMA INSTITUTE OF TECHNOLOGY

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

2023 - 2024

**BANSILAL RAMNATH AGARWAL CHARITABLE TRUST’S**

## VISHWAKARMA INSTITUTE OF TECHNOLOGY

**(An Autonomous Institute affiliated to Savitribai Phule Pune University)**

## PUNE – 411037



**CERTIFICATE**

This is to certify that the Course Project titled **“Student Record Management”** submitted by Isha Thakur(12311945), Raj Thakur(12310176), Shatananda Thakur(12310748), Prashant Thorve(12311619), Anshul Thote(12311942) Tanishq Thuse(12310237) is in partial fulfillment for the award of Degree of Bachelor of Technology in **Dept of Engineering Science and Humanities** of Vishwakarma Institute of Technology, Savitribai Phule Pune University. This project report is a record of bonafide work carried out by him under my guidance during the academic year 2023-24.

**Guide HOD, DESH**

**Prof. Lokesh Khedekar Sir Prof. Dr. C. M. Mahajan**

**Place:** VIT, Pune

**Date: 20/5/2024**

Bansilal Ramnath Agarwal Charitable Trust’s

**Vishwakarma Institute of Technology, Pune-37**

**Department of Engineering Science and Humanities**

## PROJECT SYNOPSIS

Group No : 2 Members: 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Roll No. | G.R.No. | Name of Student | Contact No. | Email ID |
| 7 | 12311860 | Isha Thakur | 9960568571 | isha.thakur23@vit.edu |
| 8 | 12311959 | Raj Thakur | 8788284481 | raj.thakur23@vit.edu |
| 9 | 12310662 | Shatananda Thakur | 9145617349 | [shatananda.thakur23@vit.edu](mailto:rohit.kamble23@vit.edu) |
| 10 | 12310963 | Prashant Thorve | 9371362622 | [prashant.thorve23@vit.edu](mailto:devanshu.kambdi23@vit.edu) |
| 11 | 12311539 | Anshul Thote | 9834150903 | [anshul.thote231@vit.edu](mailto:swamini.kanse23@vit.edu) |
| 12 | 12310532 | Tanishq Thuse | 9175807102 | [tanishq.thuse23@vit.edu](mailto:piyush.kathoke23@vit.edu) |

Academic Year : 2023-24

Project Title : Student Record Management

Project Area : Problem Solving And Programming

Internal Guide : prof. Lokesh Khedekar Sir

**Signature of Internal Guide**

## ACKNOWLEDGEMENT

Through this course project we learnt to develop a basic programming C language for management of student records from our professor, and various other resources using various concepts of programming. Also, the group members have shown the unity by working together in this course project, which really increased our team spirit. We all are very thankful to our professor and college for having this Problem solving and programming (PSAP) course in our curriculum. The way this subject is taught by our professor is really appreciating. We all know that whatever the things which we learnt about this subject will help us in our future endeavors.

# Introduction

The program for student record management in C allows the user to maintain record of many students hassle free preventing major errors in student information. In this course project we have used various concepts such as structures, pointers, conditional operators, loops, switch case, file handling and many more. In this program a set of options is displayed to the user from which the user has to choose the required option and thus the function related to that corresponding choice is run. Then the user has to enter the requires input and thus the function is executed. The program has the functions to add a student’s information, display all student’s information, searching, updating and deleting the information of a particular student using PRN, saving the data into a file and loading data from a file.

# Features provided by the C program

* It has the features of adding student information, displaying all student’s information, searching, updating, and deleting the information of a particular student using PRN, saving the data into a file and loading data from a file.
* It also prevents invalid inputs and displays error if invalid inputs are entered and the selection menu is displayed again.
* While entering the PRN in student information no duplicate entries of PRN can be made.
* Division of student must be between A to V otherwise it will display an error and the choice menu will be displayed again.
* A list of all students can be seen by selecting display all students option.
* The student data can be saved into a file and can be loaded from an already saved file thus allowing the user to enter inputs at different times.

# Program code

# #include <stdio.h>

# #include <stdlib.h>

# #include <string.h>

# #define MAX\_STUDENTS 100

# #define FILENAME "student\_data.dat"

# // Structure to represent a student

# struct Student {

# int rollNumber;

# char prn[15];

# char fullName[100];

# char branch[20];

# char division;

# float marks;

# };

# // Function to display the menu

# void displayMenu() {

# printf("\nStudent Record Management System\n");

# printf("1. Add Student\n");

# printf("2. Display All Students\n");

# printf("3. Search Student by PRN\n");

# printf("4. Update Student Information by PRN\n");

# printf("5. Delete Student\n");

# printf("6. Save Data to File\n");

# printf("7. Load Data from File\n");

# printf("8. Exit\n");

# printf("Enter your choice: ");

# }

# // Function to handle input errors

# void clearBuffer() {

# while (getchar() != '\n');

# }

# // Function to check for duplicate PRNs

# int isDuplicatePRN(struct Student students[], int numStudents, const char \*prn) {

# for (int i = 0; i < numStudents; i++) {

# if (strcmp(students[i].prn, prn) == 0) {

# return 1; // Duplicate found

# }

# }

# return 0; // No duplicate

# }

# // Function to add a new student

# void addStudent(struct Student students[], int \*numStudents) {

# if (\*numStudents == MAX\_STUDENTS) {

# printf("Maximum number of students reached.\n");

# return;

# }

# printf("Enter Roll Number: ");

# int rollNumber;

# if (scanf("%d", &rollNumber) != 1) {

# printf("Invalid input. Roll Number must be an integer.\n");

# clearBuffer();

# return;

# }

# printf("Enter PRN: ");

# char prn[15];

# scanf("%s", prn);

# if (isDuplicatePRN(students, \*numStudents, prn)) {

# printf("Student with PRN %s already exists.\n", prn);

# return;

# }

# students[\*numStudents].rollNumber = rollNumber;

# strcpy(students[\*numStudents].prn, prn);

# printf("Enter Full Name: ");

# scanf(" %[^\n]", students[\*numStudents].fullName);

# printf("Enter Branch: ");

# scanf("%s", students[\*numStudents].branch);

# printf("Enter Division (A to V): ");

# if (scanf(" %c", &students[\*numStudents].division) != 1 ||

# (students[\*numStudents].division < 'A' || students[\*numStudents].division > 'V')) {

# printf("Invalid input. Division must be A to V.\n");

# clearBuffer();

# return;

# }

# printf("Enter Marks: ");

# if (scanf("%f", &students[\*numStudents].marks) != 1) {

# printf("Invalid input. Marks must be a floating-point number.\n");

# clearBuffer();

# return;

# }

# (\*numStudents)++;

# printf("Student added successfully.\n");

# }

# // Function to display all students

# void displayAllStudents(struct Student students[], int numStudents) {

# printf("\nAll Students:\n");

# for (int i = 0; i < numStudents; i++) {

# printf("Roll Number: %d, PRN: %s, Name: %s, Branch: %s, Division: %c, Marks: %.2f\n",

# students[i].rollNumber, students[i].prn, students[i].fullName, students[i].branch, students[i].division, students[i].marks);

# }

# }

# // Function to search for a student by PRN

# void searchStudentByPRN(struct Student students[], int numStudents) {

# char prn[15];

# printf("Enter PRN to search: ");

# scanf("%s", prn);

# for (int i = 0; i < numStudents; i++) {

# if (strcmp(students[i].prn, prn) == 0) {

# printf("Student found:\n");

# printf("Roll Number: %d, PRN: %s, Name: %s, Branch: %s, Division: %c, Marks: %.2f\n",

# students[i].rollNumber, students[i].prn, students[i].fullName, students[i].branch, students[i].division, students[i].marks);

# return;

# }

# }

# printf("Student with PRN %s not found.\n", prn);

# }

# // Function to update student information by PRN

# void updateStudentByPRN(struct Student students[], int numStudents) {

# char prn[15];

# printf("Enter PRN to update: ");

# scanf("%s", prn);

# for (int i = 0; i < numStudents; i++) {

# if (strcmp(students[i].prn, prn) == 0) {

# printf("Enter new Roll Number: ");

# if (scanf("%d", &students[i].rollNumber) != 1) {

# printf("Invalid input. Roll Number must be an integer.\n");

# clearBuffer();

# return;

# }

# printf("Enter new Full Name: ");

# scanf(" %[^\n]", students[i].fullName);

# printf("Enter new Branch: ");

# scanf("%s", students[i].branch);

# printf("Enter new Division (A to V): ");

# if (scanf(" %c", &students[i].division) != 1 ||

# (students[i].division < 'A' || students[i].division > 'V')) {

# printf("Invalid input. Division must be A to V.\n");

# clearBuffer();

# return;

# }

# printf("Enter new Marks: ");

# if (scanf("%f", &students[i].marks) != 1) {

# printf("Invalid input. Marks must be a floating-point number.\n");

# clearBuffer();

# return;

# }

# printf("Student information updated successfully.\n");

# return;

# }

# }

# printf("Student with PRN %s not found.\n", prn);

# }]

# // Function to delete a student

# void deleteStudent(struct Student students[], int \*numStudents) {

# char prn[15];

# printf("Enter PRN to delete: ");

# scanf("%s", prn);

# for (int i = 0; i < \*numStudents; i++) {

# if (strcmp(students[i].prn, prn) == 0) {

# // Shift elements to remove the student

# for (int j = i; j < \*numStudents - 1; j++) {

# students[j] = students[j + 1];

# }

# (\*numStudents)--;

# printf("Student deleted successfully.\n");

# return;

# }

# }

# printf("Student with PRN %s not found.\n", prn);

# }

# // Function to save data to a file

# void saveDataToFile(struct Student students[], int numStudents) {

# FILE \*file = fopen(FILENAME, "wb");

# if (file == NULL) {

# printf("Error opening file for writing.\n");

# return;

# }

# fwrite(students, sizeof(struct Student), numStudents, file);

# fclose(file);

# printf("Data saved to file successfully.\n");

# }

# // Function to load data from a file

# void loadDataFromFile(struct Student students[], int \*numStudents) {

# FILE \*file = fopen(FILENAME, "rb");

# if (file == NULL) {

# printf("Error opening file for reading.\n");

# return;

# }

# \*numStudents = fread(students, sizeof(struct Student), MAX\_STUDENTS, file);

# if (fclose(file) != 0) {

# printf("Error closing file.\n");

# return;

# }

# printf("Data loaded from file successfully.\n");

# }

# int main() {

# struct Student students[MAX\_STUDENTS];

# int numStudents = 0;

# int choice;

# do {

# displayMenu();

# if (scanf("%d", &choice) != 1) {

# printf("Invalid input. Please enter a number.\n");

# clearBuffer();

# continue;

# }

# switch (choice) {

# case 1:

# addStudent(students, &numStudents);

# break;

# case 2:

# displayAllStudents(students, numStudents);

# break;

# case 3:

# searchStudentByPRN(students, numStudents);

# break;

# case 4:

# updateStudentByPRN(students, numStudents);

# break;

# case 5:

# deleteStudent(students, &numStudents);

# break;

# case 6:

# saveDataToFile(students, numStudents);

# break;

# case 7:

# loadDataFromFile(students, &numStudents);

# break;

# case 8:

# printf("Exiting program.\n");

# break;

# default:

# printf("Invalid choice. Please try again.\n");

# }

# } while (choice != 8);

# return 0;

# }

# Output Screenshots

# 

# 

# Conclusion

The student record management system program has enabled user to store student data in computer providing a hassle-free experience to the user. The user can also store the data and re access it afterwords by loading file in the program as concept of file handling is used in program. It also avoids invalid input from the user, also the information can be updated and deleted by taking the input as PRN and using ‘for loop’ and ‘strcpy’ checks for equal PRN and thus performs the required action. Also, the use of functions such as ‘fread’ and ‘fwrite’ has been made to load data from a file and edit it.

# 

# References

* Class Notes
* Let Us C book
* <https://www.geeksforgeeks.org/clearing-the-input-buffer-in-cc/>
* <https://youtu.be/E9HHKiwvlH8?si=HUw3J4yu_M34uBwv>