**Government Polytechnic, Malvan**

A/P- Kumbharmath, Tal- Malvan, Dist.- Sindhudurg 416606



**II-Semester**

**(Year: 2023-24)**

# MICRO PROJECT

**Course:- C Programing**

**Diploma In Computer Engineering**

**Title:** Calendar in using C

**Branch:**  Computer Engineering (CO)

**Course code:-312-303**

## Course:CO-2K

**Name Of Guide:** S. S. Haldankar

**Maharashtra State Board of Technical Education, Mumbai**

**CERTIFICATE**

This is to certify that,

of **Second** Semester of Diploma in **ComputerEngineering** of **GovernmentPolytechnic, Malvan** has completed the Micro Project satisfactorily in course **Programing in C** (312303) for the academic year 2023-24 as prescribed in the curriculum.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. no | Name | Roll No. | Enrolment no | Seat no |
| 1 | Sumit Basu Rathod | 1252 | 23110230161 |  |
| 2 | Rahul Balu Rathod | 1269 | 23110230178 | **118840** |
| 3 | Parth Pradip Veturekar | 1217 | 23110230126 |  |

Place :Malva Date :

## Subject Teacher HOD Principal

**Part – A Micro-Project Proposal**

**Title of Micro-Project :-“**Calender using C **”**

1.0 **Aims/Benefits of the Micro-Project (minimum 30-40 words**):- Benefits of a Calendar Project in C Programming Course:

* \*\*Hands-On Learning\*\*: Practical application of programming concepts.
* \*\*Problem-Solving Skills\*\*: Enhance critical thinking and problem-solving abilities.
* \*\*Understanding of Data Structures\*\*: Gain insight into data structure implementation.
* \*\*Software Engineering Principles\*\*: Apply principles like modular design and documentation. - \*\*User Interface Design\*\*: Develop UI/UX design skills.

Outcomes:

* \*\*Improved Programming Proficiency\*\*: Enhanced C programming skills.
* \*\*Stronger Problem-Solving Abilities\*\*: Better at solving complex programming challenges.
* \*\*Deeper Understanding\*\*: Gain a deeper understanding of software development.
* \*\*Portfolio Enhancement\*\*: Showcase tangible project in portfolio.
* \*\*Preparation for Future Projects\*\*: Ready for tackling more advanced programming tasks.

### 2.0 Course Outcomes Addressed :-

**The Gregorian calendar is the most widely used calendar in the world. There are leap years in the Gregorian calendar. There are 303 regular years and 97 leap years in a four-hundred-year span. The majority of people believe that every fourth year is a leap year, although this is not the case.**

### 3.0 Proposed Methodology :-

**This concludes the C programming lesson. We hope you now know how to find days-names or dates in a year, month, or week, and that you can utilize the calendar example to develop your own date/days-names functions.**

### 4.0 Action Plan(Sequence and time required for major activity):-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No | Details of activity | Planned start date | Planned Finish date | Name of  Responsible  Team  Members |
| 1 | Select Topic | 15-3-2024 | 17-3-2024 | ------- |
| 2 | Collect Information callender | 17-3-2024 | 19-3-2024 | ---- |
| 3 | We construct algorithm for program | 19-3-2024 | 22-3-2024 | ------------- |
| 4 | The implement the program and run it | 22-3-2024 | 23-3-2024 | ------ |
| 5 | Prepare Report using Plan A & B | 23-3-2024 | 1-4-2024 | -------- |

### 5.0 Resources Required (major resources such as raw material, some machining facility, software.etc.) :-

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.no | Name of Resources / Material | Specifications | Remark |
| 1 | Internet Resources | Web link | 1 |
| 2 | Programing in C  (312-303) | Book | 1 |
| 3 | Turbo-C | Application | 1 |

**Part -B Microproject report**

**Title Of Microproje**ct :- “Callender using C”

1.0 Rationale :-

Creating a calendar in a C project can serve various purposes, including:

1. \*\*Educational Purposes\*\*: Building a calendar project can help learners understand concepts such as data structures (e.g., arrays, linked lists), algorithms (e.g., date calculations), and user interface implementation in C programming.

1. \*\*Practical Applications\*\*: Calendars are widely used tools, and implementing one in C can be part of a larger project, such as a scheduling application or a personal organizer.

1. \*\*Problem-Solving Skills\*\*: Developing a calendar project involves solving various problems, such as handling leap years, calculating day of the week, and displaying dates in a user-friendly format, which can improve problemsolving skills.

1. \*\*Customization and Extension\*\*: Once the basic calendar functionality is implemented, it can be extended with additional features like event reminders, agenda views, or integration with other applications.

Overall, building a calendar project in C provides a hands-on way to reinforce programming concepts and develop practical software engineering skills.

2.0 Introduction :-

Introduction to a Calendar Project in C:

Title: “C Calendar Project: Building a Simple Calendar Application”

In today’s digital age, calendars play a crucial role in organizing our lives. From scheduling appointments to tracking important events, a calendar is an indispensable tool. In this project, we embark on a journey to develop a simple calendar application using the C programming language.

The objective of this project is to create a user-friendly calendar interface that allows users to view, navigate, and manage dates effectively. Through this endeavor, we aim to enhance our understanding of C programming concepts while implementing fundamental functionalities of a calendar system.

Key Features of the Calendar Project:

1. Displaying the current month with dates arranged in a grid format.
2. Navigating between months and years.
3. Highlighting the current date.
4. Supporting basic operations such as adding, editing, and deleting events. 5. Handling leap years and ensuring accurate date calculations.

By undertaking this project, we not only hone our programming skills but also gain insights into algorithmic problem-solving and user interface design. Additionally, this project serves as a foundation for more advanced calendar applications with additional features and functionalities.

Let’s dive into the world of C programming and embark on this exciting journey of building our very own calendar application from scratch!

3.0 Aims/Benefits of the Micro-Project :-

Benefits of a Calendar Project in C:

1. \*\*Enhanced Programming Skills\*\*: Building a calendar project in C provides an excellent opportunity to enhance your programming skills, including understanding data structures, implementing algorithms, and handling user input/output.

1. \*\*Understanding Date and Time Handling\*\*: Developing a calendar requires managing date and time functionalities accurately. Through this project, you’ll gain a deep understanding of date calculations, leap years, and day of the week calculations.

1. \*\*Improved Problem-Solving Abilities\*\*: You’ll encounter various challenges while building the calendar, such as managing different calendar views, handling user interactions, and ensuring accuracy in date calculations. Overcoming these challenges will sharpen your problem-solving abilities.

1. \*\*User Interface Design\*\*: Implementing a user-friendly interface for the calendar involves layout management, user input validation, and providing meaningful feedback. This project will help you develop skills in designing intuitive user interfaces.

1. \*\*Practical Application\*\*: Calendars are widely used in various applications, such as scheduling systems, personal organizers, and task management tools. By building a calendar project, you’ll gain practical experience that can be applied to real-world software development scenarios.

1. \*\*Customization and Extension\*\*: Once the basic calendar functionalities are implemented, you can extend the project by adding features like event reminders, recurring events, or integration with other applications. This allows for further exploration and customization according to specific requirements.

1. \*\*Portfolio Enhancement\*\*: Completing a calendar project demonstrates your proficiency in C programming and software development. It can be a valuable addition to your portfolio, showcasing your ability to tackle complex projects and deliver practical solutions.

Overall, building a calendar project in C offers numerous benefits, ranging from skill enhancement to practical application and portfolio enhancement. It’s a rewarding endeavor that can contribute significantly to your growth as a programmer.

4.0 Course Outcome Addressed :

Course Outcomes for a Calendar Project in C Programming:

1. \*\*Enhanced Programming Skills\*\*: Improved proficiency in C programming.
2. \*\*Stronger Problem-Solving Abilities\*\*: Development of robust problemsolving skills.
3. \*\*Deeper Understanding\*\*: Increased comprehension of software development principles.
4. \*\*Portfolio Development\*\*: Creation of a tangible project for portfolio enhancement.
5. \*\*Preparedness for Future Projects\*\*: Better readiness for tackling complex programming tasks.

5.0 Actual methodology followed :-

Methodology for a Calendar Project in C Programming (Short):

1. \*\*Requirement Analysis\*\*: Define functionalities.
2. \*\*Design\*\*: Plan data structures and UI.
3. \*\*Implementation\*\*: Code date calculations, UI, and user interactions.
4. \*\*Testing\*\*: Test for functionality and correctness.
5. \*\*Debugging\*\*: Identify and fix bugs.
6. \*\*Documentation\*\*: Document code and dependencies.
7. \*\*User Feedback\*\*: Gather feedback for improvements.
8. \*\*Deployment\*\*: Deploy the application for use.

# Code :-

#include <stdio.h>

#include <stdlib.h>

int isLeapYear( int year ); int leapYears( int year ); int todayOf( int y, int m, int d); long days( int y, int m, int d); void calendar(int y, int m); int getDayNumber(int d,int m,int y); char \*getName(int day); void flush() {

int c;

while ((c = getchar()) != '\n' && c != EOF);

}

typedef struct { int day; int month; int year; char note[255];

} Note;

int main() { int year,month, day; char choice; Note note;

FILE \*fp;

fp = fopen("note.bin", "r"); if (fp == NULL) {

fp = fopen("note.bin", "w");

}

fclose(fp);

while(1)

{ printf("1. Find the day\n"); printf("2. Print calendar of a month\n"); printf("3. Add Note\n"); printf("4. Exit\n"); printf("Enter your choice: "); scanf("\n%c", &choice); switch(choice) { case '1':

printf("Enter the day, month and year: "); scanf("%d %d %d", &day, &month, &year);

printf("The day is : %s\n", getName(getDayNumber(day, month, year))); break; case '2':

printf("Enter the month and year: "); scanf("%d %d", &month, &year);

printf("Please enter 's' to see the notes\n Press any other key to continue\n");

calendar(year, month); break; case '3':

printf("Enter the day, month and year: "); scanf("%d %d %d", &note.day, &note.month, &note.year); flush();

printf("Enter the note: "); fgets(note.note, 255, stdin); fp = fopen("note.bin", "a+");

if (fp == NULL) {

printf("File note.bin can not be opened\n");

exit(1);

}

fwrite(&note, sizeof(Note), 1, fp);

printf("Note added sucessfully\n");

fclose(fp); break; case '4': printf("Bye!!"); exit(0); break; default:

printf("Not a valid option\n"); break;

}

}

}

int isLeapYear( int y ){ return(y % 400 == 0) || ((y % 4 == 0) && (y % 100 != 0));

}

int leapYears( int y ){ return y/4 - y/100 + y/400;

}

int todayOf( int y, int m, int d) { static int DayOfMonth[] =

{ -1,0,31,59,90,120,151,181,212,243,273,304,334}; return DayOfMonth[m] + d + ((m>2 && isLeapYear(y))? 1 : 0);

}

long days( int y, int m, int d){ int lastYear; lastYear = y - 1;

return 365L \* lastYear + leapYears(lastYear) + todayOf(y,m,d);

}

void calendar(int y, int m){

FILE \*fp; Note\* notes, note; int len, j, hasNote = 0; char choice; const char \*NameOfMonth[] = { NULL/\*dummp\*/,

"January", "February", "March", "April", "May", "June",

"July", "August", "September", "October", "November", "December"

};

char Week[] = "Su Mo Tu We Th Fr Sa"; int DayOfMonth[] =

{ -1,31,28,31,30,31,30,31,31,30,31,30,31 }; int weekOfTopDay; int i,day;

weekOfTopDay = days(y, m, 1) % 7;

fp = fopen("note.bin", "rb"); if (fp == NULL) {

printf("Couldn't read notes\n");

} len = 0; while(fread(&note, sizeof(Note), 1, fp)) { if (note.year == y && note.month == m) { len++;

}

} rewind(fp); j = 0; notes = (Note\*) malloc (sizeof(Note) \* len); while(fread(&note, sizeof(Note), 1, fp)) { if (note.year == y && note.month == m) { notes[j] = note; j++;

}

}

fclose(fp);

if(isLeapYear(y)) DayOfMonth[2] = 29;

printf("\n %s %d\n%s\n", NameOfMonth[m], y, Week); for(i=0;i<weekOfTopDay;i++)

printf(" ");

for(i=weekOfTopDay,day=1;day <= DayOfMonth[m];i++,day++){ hasNote = 0; for (j = 0; j < len; j++) { if (notes[j].day == day) { printf("|%2d| ",day); hasNote = 1; break;

}

}

if (hasNote == 0) { printf("%2d ",day);

} if(i % 7 == 6) printf("\n");

} printf("\n"); scanf("\n%c", &choice); if (choice == 's') { printf("Here are list of notes for %d %d\n", m, y); for (j = 0; j < len; j++) { printf("%d: %s\n", notes[j].day, notes[j].note);

}

} else { return;

} }

int getDayNumber(int d, int m, int y){ //retuns the day number static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4}; y -= m < 3;

return (y + y/4 - y/100 + y/400 + t[m-1] + d) % 7;

}

char \*getName(int day){ //returns the name of the day switch(day){ case 0 :return("Sunday"); case 1 :return("Monday");

case 2 :return("Tuesday"); case 3 :return("Wednesday"); case 4 :return("Thursday"); case 5 :return("Friday"); case 6 :return("Saturday"); default:return("Error: Invalid Argument Passed");

}

}

# Output:-



