

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in Information Technology

By
PRAGADEESWARI M

38120060



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU

JANUARY-2021

DECLARATION

I hereby declare that the Project Report entitled "**CALENDER APPLICATION**" done by me under the guidance of Dr. SIVASANGARIA M.E., Ph.D., at Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:

Place:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing** and **Dr. R. SUBHASHINI M.E., Ph.D., Head of the Department, Department of Information Technology** for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project **Dr. SIVASANGARIA M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

ABSTRACT

Calendar application is an android application which accesses the calendars of multiple users, for instance members of a family or staff in a particular department and will pull out the events from every calendar. Hence, everyone in the group can see others' events on one screen in customized format. It also helps the user to create, edit, and delete events of his/her calendar. Additionally, this application will use an efficient strategy to update data only if there is any change in the events, making it faster. There are many other applications in the Play store but they cannot get rid of redundant events and reminders of events belonging to other users. This causes a lot of disturbance to the user. On the contrary, Calendar application will use an intelligent algorithm to compare the events in many ways based on the time, name, and place of event. Furthermore, this application will only generate reminders for the user events.

CHAPTER NO	TITLE NAME	PAGE NO
	ABSTRACT	7
	LIST OF FIGURES	9
	LIST OF ABBREVIATION	10
1	INTRODUCTION	11
	1.1 ABOUT	
2	REQUIREMENT SPECIFICATION	12
	2.1 HARDWARE REQUIREMENT	
	2.2 SOFTWARE REQUIREMENT	
3	DATASRUCTURE USED	13
4	ALGORITHM	14
5	PROJECT DEFENETION	15
6	COCLUSION & FEATURES	16
	6.1 COCLUSION	
	6.2 FEATURES	
7	SOURCE CODE	18
8	OUTPUT	23

**LIST OF
FIGURES:**

SNO	FIGURE NAME	PAGE NO
1	Fig 1.1	13
2	Fig 1.2	14
3	Fig 2 output	23

INTRODUCTION

This mini project on **Calendar in C** programming language is a console application without graphics. To make the calendar look colorful, many windows properties have been used in this project. Besides the color used in backgrounds, the days of the month are white and the vacations (Sundays) are indicated using the red foreground color.

In this mini project, you can find out the day corresponding to a given date and view the days and dates corresponding to a particular month+year. The source code is not that long, over 400 lines. It is compiled in Code::Blocks IDE with GCC compiler.

To help you understand the calendar mini project better, there are lots of comments within the source code. This project is aimed to teach you "how to make Calendar using C". Click on the link below to directly download the source code along with the executable file.

1.1 About Calendar in C:

Basically three operations can be done in this calendar application. To find out the day corresponding to a given date, the date, month and year are asked. You can list the days and dates of any month of any year. For example, entering 04 2014 (April 2014) will give you an output as shown in the screenshot in this post.

You can navigate the months using arrow keys, or press 'n' and 'p' keys to view the next and previous months respectively. The third feature of this C mini project on Calendar application utilizes file handling. With this feature, you can add important notes with corresponding dates.

The functions used in the source code are simple and easy to understand.

The ones listed below have been used to produce background with color effects. They are described in the source code with comments.

- `void SetColor(int ForgC)`
- `void ClearConsoleToColors(int ForgC, int BackC)`
- `void SetColorAndBackground(int ForgC, int BackC)`

`void gotoxy (int x, int y)` – You need to understand this function as it is an important one used in this Calendar in C language. You can find this function used in many C projects. This function allows you to print text in any place of screen. Using this function in Code::Blocks requires coding, but it can be directly used in Turbo C. Here is a code for this function in Code::Blocks.

Implementation Guidelines

- Starting with IDE, you can use IDE like turbo c, dev c or code blocks.
- So we have to use pointer array to mention month name accordingly.
- Use for loop to print the 12-month headings.
- Also, use printf statement to display the weekdays in every month. It should be coded in the same for loop which was previously used for the months.
- Till now it will display the headings of 12 months including weekdays in it.

This program shows calendar of
a given month. Enter month, year...format is mm-yyyy.

Enter mm-yyyy: 06-2014

June 2014

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

(*) Use Left,Right,Up & Down Arrow.

(*) Press I for New Month & Year.

(*) Press P for Print to File.

(*) Press ESC for Exit.

Created by: Debabrata Das [debabrata.dd@gmail.com]



May 2014

SUN	MON	TUE	WED	THU	FRI	SAT
-	-	-	-	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

(*) Use Left,Right,Up & Down Arrow.

(*) Press I for New Month & Year.

(*) Press P for Print to File.

(*) Press ESC for Exit.

Created by: Debabrata Das [debabrata.dd@gmail.com]

REQUIREMENT SPECIFICATION

2.1 HARDWARE REQUIREMENTS

Processor:Core i3/ i5/ i7

RAM : 128 MB

Hard Disk : 20GB

Monitor : 15” Color monitor

Key Board : 104 Keys

2.2 SOFTWARE REQUIREMENTS

Operating System : Windows 7/8.

Language : C

Software : Turbo Cpp compiler

3. DATA STRUCTURE USED

STACK:

In computer science, a *stack* is an abstract data type that serves as a collection of elements, with two principal operations:

- **push**, which adds an element to the collection and
- **pop**, which removes the most recently added element that was not yet removed.

The order in which elements come off a stack gives rise to its alternative name, **LIFO (last in, first out)** or **FILO(first in , last out)**. Additionally, a peek operation may give access to the top without modifying the stack.^[1] The name "stack" for this type of structure comes from the analogy to a set of physical items stacked on top of each other, which makes it easy to take an item off the top of the stack, while getting to an item deeper in the stack may require taking off multiple other items first.^[2]

Considered as a linear data structure, or more abstractly a sequential collection, the push and pop operations occur only at one end of the structure, referred to as the *top* of the stack. This makes it possible to implement a stack as a singly linked list and a pointer to the top element. A stack may be implemented to have a bounded capacity. If the stack is full and does not contain enough space to accept an entity to be pushed, the stack is then considered to be in an **overflow** state. The pop operation removes an item from the top of the stack.

A stack is needed to implement depth-first search.

4. ALGORITHM

BEGIN

while(1)

BEGIN

printf("1. Find the day\n");

printf("2. Print calendar of a month\n");

printf("3. Add Note\n");

printf("4. Exit\n");

switch(choice)

BEGIN

case'1':

scanf(&day, &month, &year);

break;

case'2':

scanf(&month, &year);

calendar(year, month);

break;

case'3':

scanf(¬e.day, ¬e.month, ¬e.year);

flush();

printf("Enter the note: ");

fgets(note.note, 255, stdin);

fp = fopen("note.bin", "a+");

if (fp == NULL)

BEGIN

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

END

fwrite(¬e, sizeof(Note), 1, fp);

printf("Note added sucessfully\n");

fclose(fp);

```
break;
case'4': printf("Bye!!");
break;
default:printf("Not a valid option\n");
break;
END
END
END
```

5. PROJECT DESCRIPTION

- 1.It displays a nicely formatted calendar of every month.
2. You can find the day by entering the day, month and year. For example, if you enter day = 10,month = 03 and year = 1991, it gives you the day 'Sunday'.
3. You can add the note for a particular day.
4. If the given month has a note in it, it will display in that day.

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Our project is a humble venture to satisfy the need of customer who use the calendar application. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the customer as well as organization.

The objective of software planning is to provide a frame work that enables the user to view the calendar made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

5.2 FUTURE WORK

1. It displays a nicely formatted calendar of every month.
2. You can find the day by entering the day, month and year. For example, if you enter day = 10, month = 03 and year = 1991, it gives you the day 'Sunday'.
3. You can add the note for a particular day.
4. If the given month has a note in it, it will display || in that day.

5.3 REFERENCES

<http://www.tutorialspoint.com/cplusplus/>

Data Structures And Algorithm Analysis In C++ by Mark Allen Weiss

Introduction To Algorithm –Third Edition By Cormen.

http://scanftree.com/Data_Structure/

6.SOURCE CODE

```
#include <stdio.h>
#include <stdlib.h>
intisLeapYear( int year );      /* True if leap year */
intleapYears( int year );      /* The number of leap year */
inttodayOf( intyr, intmnth, int d); /* The number of days since the beginning of the year */
long days( intyr, intmnth, int d); /* Total number of days */
void calendar(intyr, intmnth);  /* display calendar at m y */
intgetDayNumber(intd,intmnth,intyr);
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(intargc, char* argv[]){
    intyear,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.bincan not be opened\n");
            }
        case '4':
            break;
    }
}

```

```

    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;
}
}
return 0;
}

```

```

intisLeapYear( intyr ){
    return(yr % 400 == 0) || ((yr % 4 == 0) && (yr % 100 != 0));
}

```

```

intleapYears( intyr ){
    returnyr/4 - yr/100 + yr/400;
}

```

```

inttodayOf( intyr, intmnth, int d) {
    staticintDayOfMonth[] =
        { -1,0,31,59,90,120,151,181,212,243,273,304,334};
    returnDayOfMonth[mnth] + d + ((mnth>2 &&isLeapYear(yr))? 1 : 0);
}

```

```

long days( intyr, intmnth, int d){
    intlastYear;

```

```

    lastYear = yr - 1;
    return 365L * lastYear + leapYears(lastYear) + todayOf(yr,mnth,d);
}

void calendar(intyr, intmnth){
    FILE *fp;
    Note* notes, note;
    intlen, j, hasNote = 0;
    char choice;

    const char *NameOfMonth[] = { NULL/*dummp*/,
        "January", "February", "March", "April", "May", "June",
        "July", "August", "September", "October", "November", "December"
    };
    char Week[] = "Sun Mon Tue Wed Thur Fri Sat";
    intDayOfMonth[] = { -1,31,28,31,30,31,30,31,31,30,31,30,31 };
    intweekOfTopDay;
    inti,day;

    weekOfTopDay = days(yr, mnth, 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 == yr&&note.month == mnth) {
            len++;
        }
    }
}

```

```

rewind(fp);
j = 0;
notes = (Note*) malloc (sizeof(Note) * len);
while(fread(&note, sizeof(Note), 1, fp)) {
    if (note.year == yr&&note.month == mnth) {
        notes[j] = note;
        j++;
    }
}
fclose(fp);
if(isLeapYear(yr))
    DayOfMonth[2] = 29;
printf("\n    %s %d\n%s\n", NameOfMonth[mnth], yr, Week);

for(i=0;i<weekOfTopDay;i++)
printf(" ");
for(i=weekOfTopDay,day=1;day <= DayOfMonth[mnth];i++,day++){
    hasNote = 0;
    for (j = 0; j <len; j++) {
        if (notes[j].day == day) {
            printf("|%2d| ",day);
            hasNote = 1;
        }
    }
    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", mnth, yr);
    for (j = 0; j < len; j++) {
        printf("%d: %s\n", notes[j].day, notes[j].note);
    }
} else {
    return;
}
}

int getDayNumber(int d, int mnth, int yr) { //returns the day number
    static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};
    yr -= mnth < 3;
    return (yr + yr/4 - yr/100 + yr/400 + t[mnth-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:

```
3
1991
The day is : Sunday
1. Find the day
2. Print calendar of a month
3. Add Note
4. Exit
Enter your choice: 3
Enter the day, month and year: 16
10
2020
Enter the note: Note added sucessfully
1. Find the day
2. Print calendar of a month
3. Add Note
4. Exit
Enter your choice: 4
Bye!!
1. Find the day
2. Print calendar of a month
3. Add Note
4. Exit
Enter your choice: 2
Enter the month and year: 10
2020
Please enter 's' to see the notes
Press any other key to continue

October2020
Sun  Mon  Tue  Wed  Thur  Fri  Sat
    1   2   3
 4   5   6   7   8   9  10
11  12  13  14  15  16  17
18  19  20  21  22  23  24
25  26  27  28  29  30  31

1
1. Find the day
2. Print calendar of a month
3. Add Note
4. Exit
Enter your choice: 1
Enter the day, month and year: 10
3
1991
The day is : Sunday
1. Find the day
2. Print calendar of a month
3. Add Note
4. Exit
Enter your choice: 3
Enter the day, month and year: 16
10
```