National University
of computer and emerging sciences

Programming fundamentals Project

Project Title:

Timetable Management System

Group Members:

1) Ayesha Zia*(20K-0414)*
2) Syeda Ravia Ejaz*(20K-0343)*

➢ Introduction:

Timetable Management System is designed mainly for two users: Students and Teachers. The program aims to make storing and obtaining different types of information like courses, classes, attendance, and assignments more efficient and less time consuming.

The features vary for both the users and are as following:

Students:

1) Viewing timetable of classes
2) Getting directions to the class
3) Getting details of upcoming assignment and quizzes
4) Obtaining details for different courses offer by the institute

Teachers:

1) Marking attendance of their class
2) Checking free available slots on a particular day
3) Posting assignment/Quizzes for students

➢ Background:

information (attendance, posting assignments/quizzes) and obtaining the desired information(timetables, pending assignments, course details and free slots) is often inefficient and time consuming due to less automation and more manual availability of data. The timetable management system is designed with these problems faced by students and teachers in mind. The program aims to automate the process of taking records and getting the desired information.

➢ Problem Analysis:

The project targets multiple problems faced by students and teachers when obtaining and storing information manually. The analysis and countering of problems via the program is as following:

1) **Timetable**

   Analysis:

   The timetable uploaded by the institute is a common excel file with master timetable. The student has to go through toiling to view his timetable and relevant classes as the master timetable has details of all sections, most of which is irrelevant.

   Solution:

   Through the Timetable Management System, students only need to enter the day and time in the program, and they will get info regarding the class they have at that particular time.

2) **Directions**

   Analysis:

   Students, specially freshmen, tend to get confused when finding their classes where the lecture is to be delivered. This is too due to different buildings(CS/EE), number of rooms and similarity in name of room(CR-10,E_4 etc.).

   Solution:

The program will ask the student if they want to know the directions to class after they view their timetable for any particular day/time. The student can the enter the room name and the directions to that room will be displayed.

3) **Details of upcoming assignments and quizzes**

Analysis:

If manually noted, students often tend to forget about the pending assignments due to the bulk of assignments of different courses.

Solution:

The student can simply enter the day for which he wants to know the pending assignments/quizzes posted by the teacher. In case a student misses a lecture, he can still know if the teacher gave an assignment on that day.

4) **Course details**

Analysis:

To get details of a particular courses, students have to go through the lengthy prospectus which contains the details of all the departments and courses in one place.

Solution:

Instead of manually looking up, the student can simply enter the course name in the Timetable Management System. The program will provide the relevant details(i.e.: Course instructor, credit hours and course code).

5) **Marking attendance**

Analysis:

For teachers, marking attendance manually is easy but can be time consuming as they have to cross check for sections and student names.

Solution:

In the program, the teacher can enter the section for which the teacher has to mark attendance. After the teacher has input the attendance, it is stored in the form of a text file in the respective folder mentioned in the code. This way, for each day, the attendance is stored in form of text files in the folder and replaces manual attendance marking.

As the program is generic, the class strength has to be specified in the program. However, the names of students in each section can be integrated in the program through text filing.

6) **Checking free slots available**

Analysis:

If a teacher has to find their free slots for a particular day, they have to look through the master timetable which has details of all days, time slots and teachers.

Solution:

Through the Timetable Management System, the teacher only has to input the day and the free slots for that particular day will be displayed on screen.

7) **Posting assignment/quizzes**

Analysis:

Making announcement in class for assigning a task or quiz is often inefficient as some students might miss the lecture on a particular day.

Solution:

The teacher just has to input the day on which they want to assign the quiz and then they will be allowed to enter details of the assignment/quiz. The assignment details will then be stored through filing in a text file that can be viewed by the students(See Problem Analysis 2).

(SEE "RESULTS" BELOW FOR OUTPUTS DISPLAY)

➢ Methodology:

1) LIBRARIES:

The program uses the following built-in libraries of DEV C++:

- #include<stdio.h>
- #include<windows.h>
- #include"mmsystem.h"
- #include<stdlib.h>
- #include<string.h>

2) TOOLS:

The program uses the following tools taught in the Programming Fundamentals course:

- Switch statements
- Conditional statements(If/else/else-if)
- For, while loops, break, continue
- Structures
- Functions
- Pointers
- Filing
- Strings
- One dimensional and two-dimensional arrays

3) FUNCTIONS:

The program uses the following functions:

- The program uses the following function to integrate sound API(application programming interference) in the timetable management system:
  ```
  PlaySound("C:/WINDOWS/Media/ding.wav", NULL, SND_FILENAME)
  ```

- The program uses the following function in the output stream to clear the buffer output and then move the data that is buffered to console or disk:
  ```
  fflush(stdin);
  ```

- The program uses the following function to clear the screen and move the cursor to the next line of the source code:

```
system("CLS");
```

4) COMMANDS:

The program runs with audio outputs that run successfully when the following commands are added in the compiler options in DEV C++:

- Compiler command: -std=c99
- Linker command: -static-libgcc -lwinmm

➢ Implementation:

The tools mentioned in methodology were implemented throughout the program in the following manner:

1) **Switch statements:** The switch statements were used when the program takes input from user.

```
    switch(choice)
    {
        case 1:{
           teacher();
           break;
        }
        case 2:{
            student();
            break;
        }
        case 3:{
            system("CLS");
            printf("\n\n\n\n\n\n\t\t\tThank you for using time table management system!!!!");
            printf("\n\n\t\t\tProject designed by:\n\t\t\t\t* Ayesha Zia(20k-0414)\n\t\t\t\t* Syeda Ravia Ejaz(20k-0246)\n\n\n\n\n\n");
            break;
        }
    }
}
```

2) **Conditional statements(if/else):** IF statements check through the user input that if a condition is true or not. If not, the ELSE statement specifies what the program should do otherwise.

```
    if(x == 1)
    {
        student();
    }
    else if(x == 2)
    {
        system("CLS");
        printf("\n\n\n\n\n\n\t\t\tThank you for using time table management system!!!!");
        printf("\n\n\t\t\tProject designed by:\n\t\t\t\t* Ayesha Zia(20k-0414)\n\t\t\t\t* Syeda Ravia Ejaz(20k-0246)\n\n\n\n\n\n");
    }
}
else{
system("CLS");
printf("\n\n\n\t\t      <><><><><> SATURDAY <><><><><>\n\n");
printf("\t\t\t* Saturday at 11 AM:\n ");
```

3) **Loops:** The loops were used to when the program has to take input from user until the desired limited entered by user is reached.

```
for(i=0;i<=7;i++)
{
    printf("%c", Monday[0][i]);
}
printf("\n\t\t\t* Classroom = CR-4");
printf("\n\t\t\t* Course instructor = Sir Waqar Ahmed");
```

4) **Structures:** Structures were used to store similar data type in a grouped form.

```
struct students{
    char attend[5];
};
```

5) **Functions:** Functions were used to avoid writing the same code multiple times. Functions helped in shortening the source code and dividing he code in modular functions makes it easy to write and read the source code.

```
void teacher();
int student();
void classes();
void direction();
void assignment();
void course_details();
void attendance();
void free_slots();
void quiz();
```

6) **Pointers and filing:** Filing was used to store data entered by the user in the system like attendance.

```
FILE *fptr;
 fptr=(fopen("d:\\attendance.txt","w"));
 if(fptr==NULL) {
    printf("Error!");
      }
for(i=0;i<str;i++){
printf("Student %d : ",i+1);
scanf("%s",&st[i].attend);
fflush(stdin);
fprintf(fptr,"\nStudent: %d \n Present/Absent=%s \n",i+1,st[i].attend);
}
```

Functions were also used to read data stored in a text file in the system of user(example: directions).

```
    FILE *ptr;
    ptr = fopen("CR-4.txt", "r");
    if(ptr==NULL){
        printf("File not opened!");
        exit(1);
    }
    printf("\n\n\t\t     <><><><><><>CLASSROOM-4<><><><><><>");
printf("\n\n\t\t\t\t*");
for(ch==getc(ptr);ch!=EOF;ch=getc(ptr))
{
    if(ch=='\n')
        printf("\n\t\t\t* ");
    else
        printf("%c", ch);
}
fclose(ptr);
```

7) **Strings:** Strings were used when the program has to take input from user like the of the day and string function like strlen() was used to specify the length of string.

```
scanf("%s", &day);
var2 = strlen(day);
if(strlen(day)==6)
{
```

8) **Arrays:** One-dimensional and two-dimensional arrays were used to store and group together similar elements like

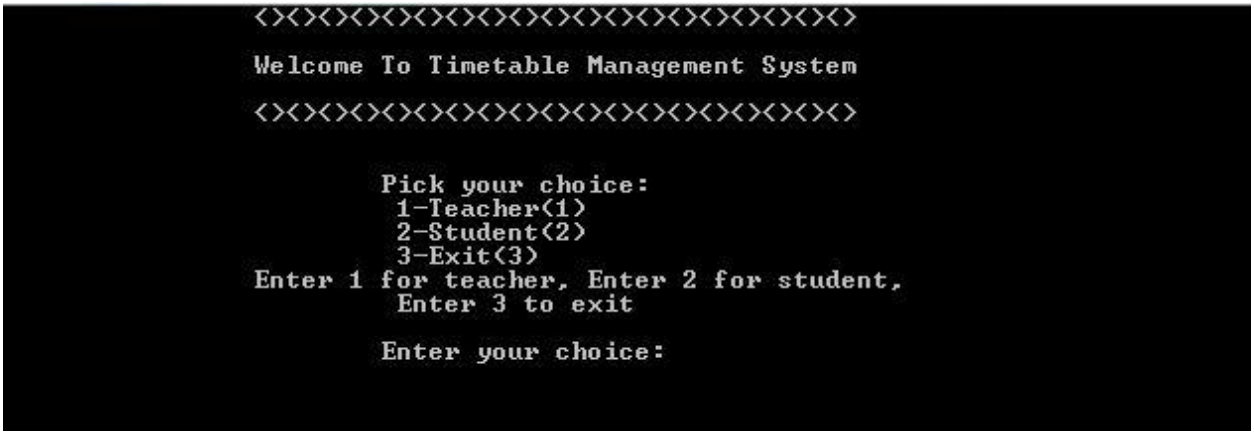
```
char day[15];
int Time[8] = { 8,9,10,11,12,2,3,4 };
```

```
char Monday[8][15] = { "Physics", "Calculus", "IRS", "Break", "Programming", "ICT", "ICT", "ICT" };
char Tuesday[8][15] = { "English", "Break", "Calculus", "Physics", "Break", "Programming", "Programming", "Programming"};
char Wednesday[8][15] = { "Physics", "IRS", "English", "Break", "Programming", "Calculus", "Calculus", "ICT" };
char Thursday[8][15] = { "Calculus", "Physics", "Physics", "Break", "IRS", "English", "Programming", "ICT" };
char Friday[8][15] = { "ICT", "Calculus", "Break", "Calculus", "IRS", "English", "Programming", "Physics" };
char Saturday[8][15] = { "Physics", "Break", "Programming", "IRS", "English", "IRS", "Calculus", "ICT" };
```

➢ Results:

The results of the program are as following:

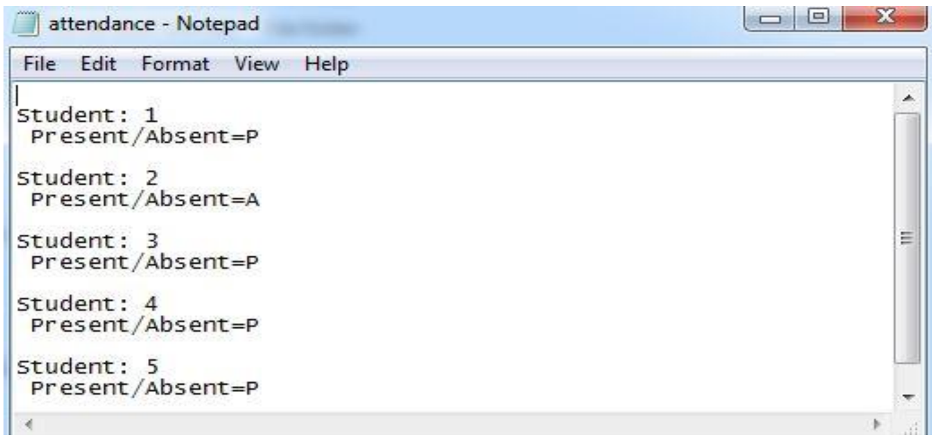
Main screen



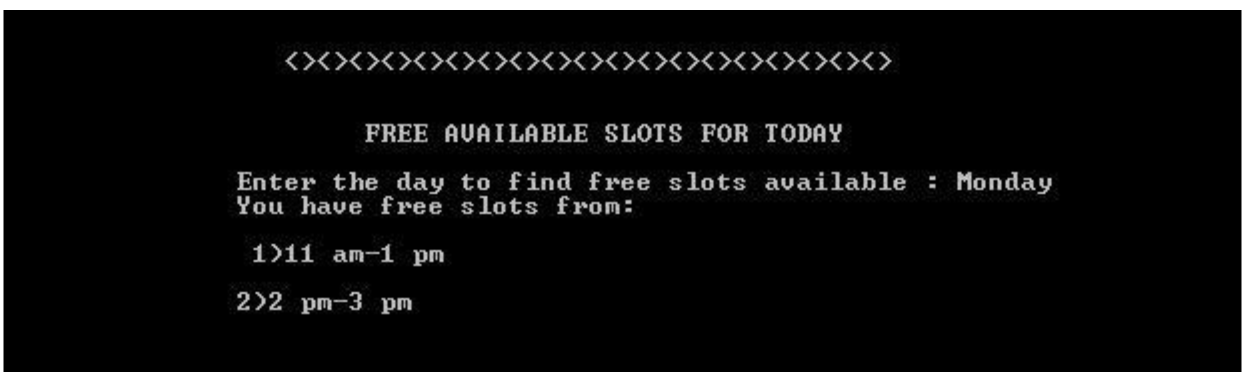
Teacher Module Main Screen

Marking Attendance

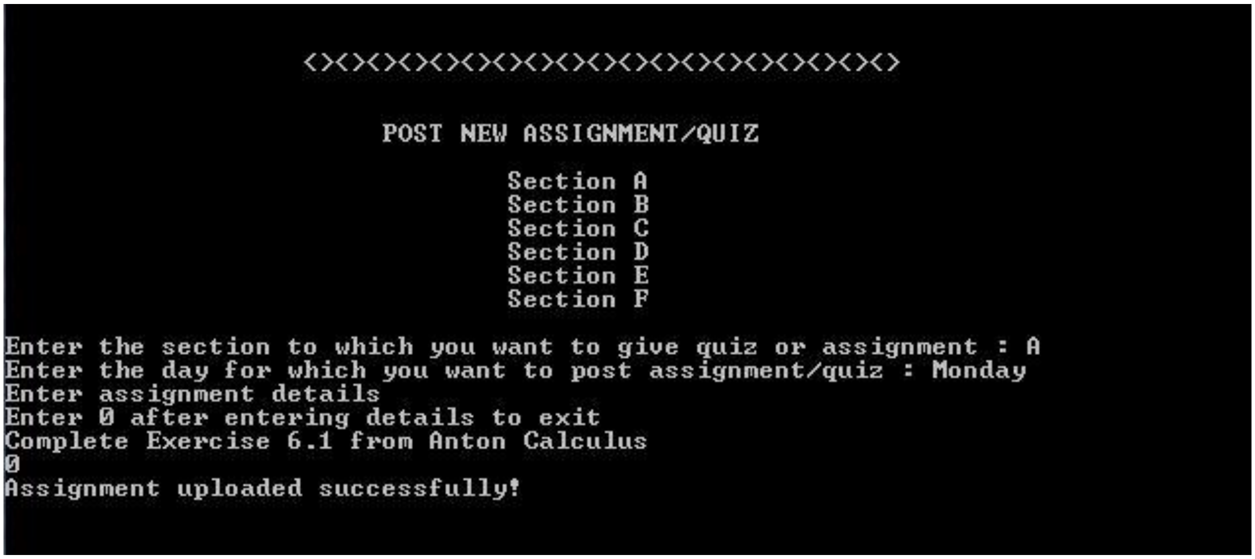

<><><><><><><><><><><><><><><><><><><>
ATTENDANCE
Mark P for present, Mark A for absent
Enter the class strength : 5
Student 1 : P
Student 2 : A
Student 3 : P
Student 4 : P
Student 5 :

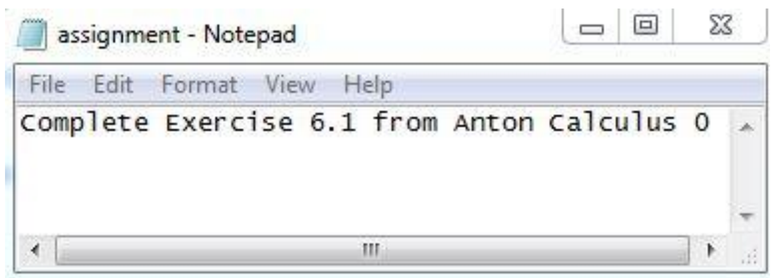

attendance - Notepad
File Edit Format View Help
Student: 1
Present/Absent=P
Student: 2
Present/Absent=A
Student: 3
Present/Absent=P
Student: 4
Present/Absent=P
Student: 5
Present/Absent=P

Free slots

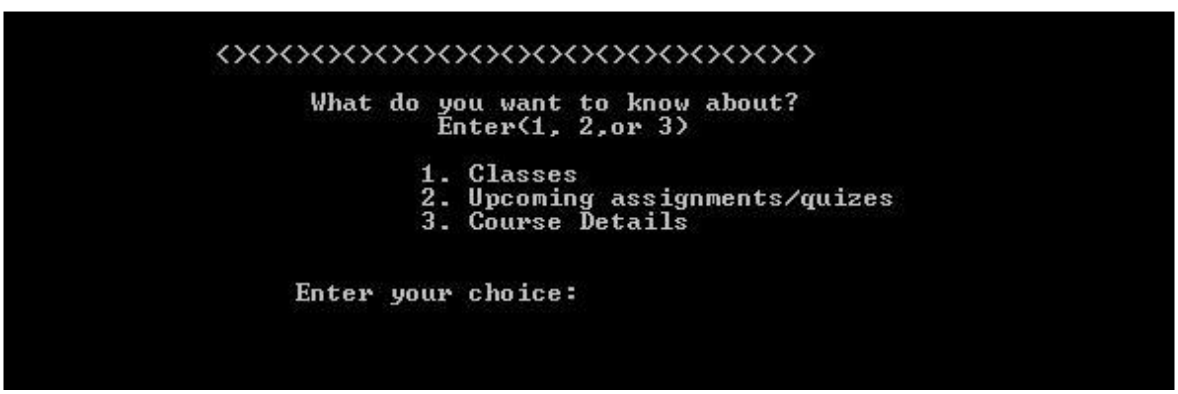

<><><><><><><><><><><><><><><><><><><>
FREE AVAILABLE SLOTS FOR TODAY
Enter the day to find free slots available : Monday
You have free slots from:
1)11 am-1 pm
2)2 pm-3 pm

Posting Assignment/Quiz





Student Module Main Screen



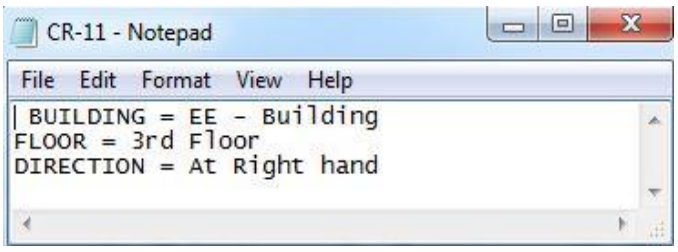
Classes

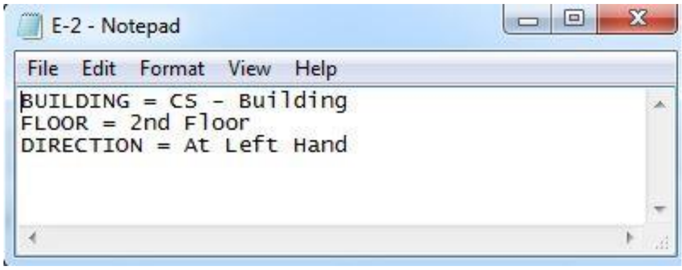
```
<><><><><> CLASSES <><><><><>

* Enter the time of class: 8
* Enter the day of the class: Monday
```

```
<><><><><> MONDAY <><><><><>

    * Monday at 8 AM:
    * Class = Physics
    * Classroom = CR-4
    * Course instructor = Sir Waqar Ahmed

Do you want to know the direction of the class?
      Enter(1)if yes or (2)if no

       1. YES
       2. No

Enter your choice:
```

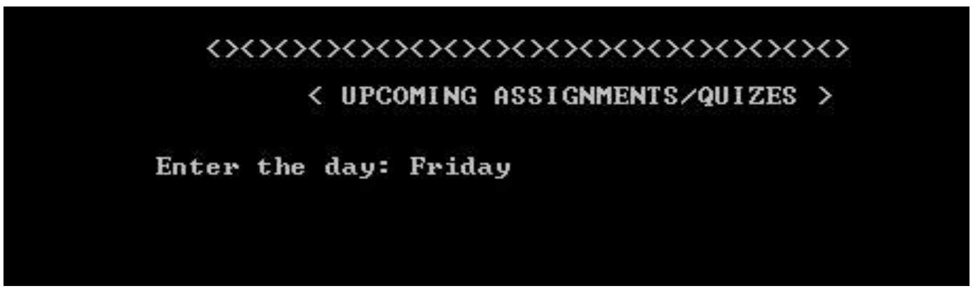
DIRECTIONS TO THE CLASS

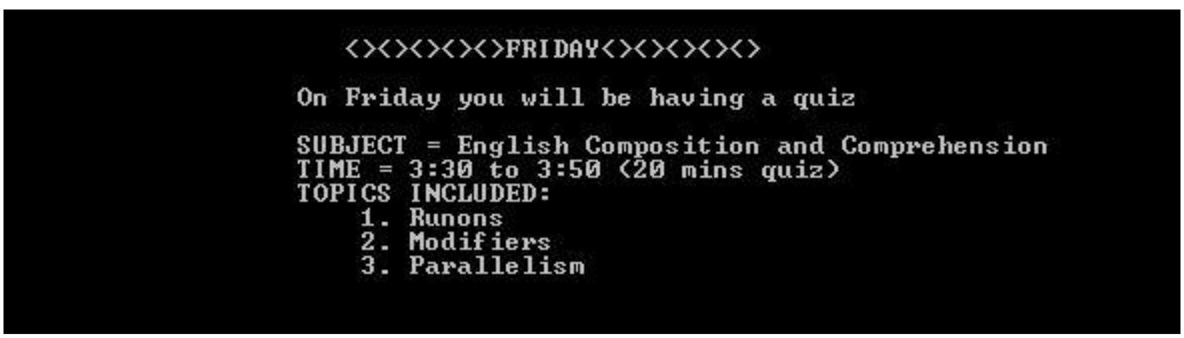
```
<><><><><><><> DIRECTIONS OF CLASS <><><><><><><>

     <><><><><><>CLASSROOM-7<><><><><><>

        * BUILDING = EE - Building
        * FLOOR = 2nd Floor
        * DIRECTION = At Right Hand
```



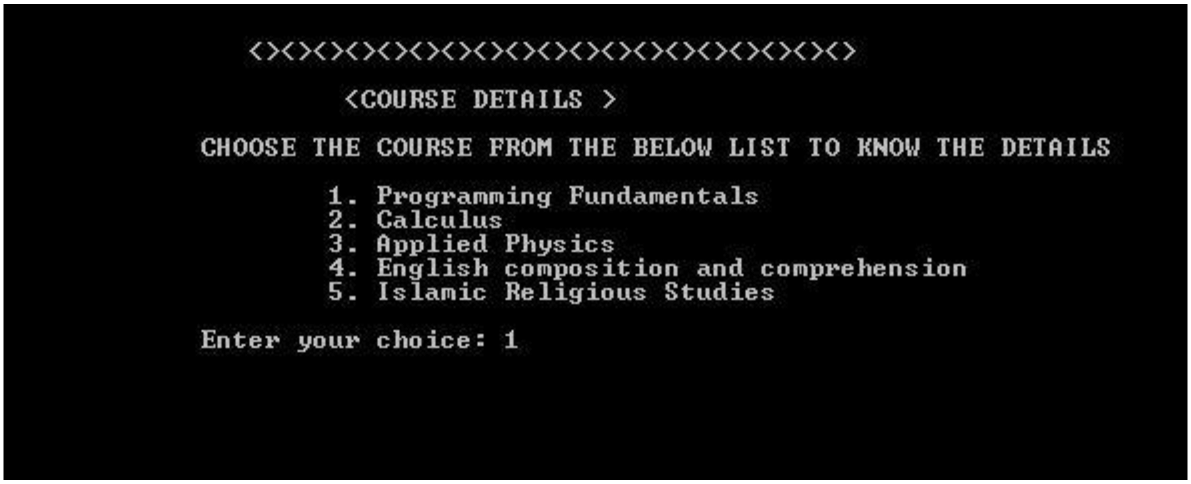


Upcoming Assignment/Quizzes

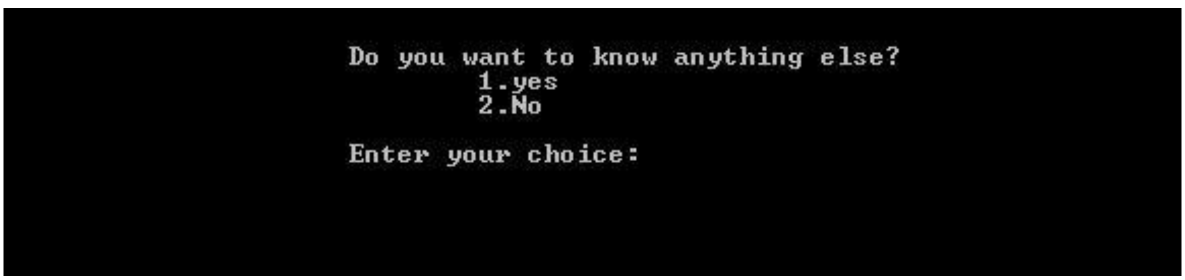




Course Details



IF THE USER WANTS TO KNOW ANYTHING ELSE



LAST SCREEN