Course Management System

Specifications:

- **Variables:** Course ID, name, duration, and credits.
- **Static & Const:** Static variable for total courses; const for maximum number of courses.
- **Switch Case:** Menu for adding, updating, and viewing courses.
- **Looping Statements:** Loop through course list.
- **Pointers:** Pointer for dynamic course name allocation.
- **Functions:** Separate functions for each course operation.
- **Arrays:** Store course details.
- **Structures:** Structure for course information.
- **Nested Structures:** Nested structures for course and instructor details.
- **Unions:** Union for course types.
- **Nested Unions:** Nested union for different instructional formats.
- **Output Expectations:** Display course list with details.

Menu Example:

1. Add Course

2. Update Course

3. View Courses

4. Exit

```
#include<stdio.h>

#include<stdlib.h>

#define MAX_COURSE 5


struct Instructor{

  char course[20];

  char instructor[20];

};
```

```
struct Course{
  int courseID;
  char name[30];
  float duration;
  float credits;
  struct Instructor instructorDetails;
};


int static total_courses=0;

void addCourse(struct Course *details, int *total_courses);
void updateCourse(struct Course *details, int total_courses);
void viewCourse(struct Course *details, int total_courses);

int main(){

    struct Course *details=(struct Course *)malloc(MAX_COURSE*sizeof(struct Instructor));
    if(details==NULL){
        printf("Memory allocation failed");
        return 1;
    }

    int choice;
    do{
```

```
    printf("\nCourse management system!\n");
printf("Enter the choice: \n");
printf("1.Adding new Course \n2.Updating course \n3.View entire courses\n4.Exit");
scanf("%d",&choice);

switch (choice){
    case 1:
    addCourse(details,&total_courses);
    break;

    case 2:
    updateCourse(details,total_courses);
    break;

    case 3:
    viewCourse(details,total_courses);
    break;

    case 4:
    printf("Exit!");
    break;

    default:
    printf("Invalid Choice");
}
```

```
    }while(choice!=4);

    free(details);
    return 0;

}

void updateCourse(struct Course *details, int total_courses){
     if(total_courses==0){
        printf("\nNo added Course\n");
        return;
    }
    int searchID;
    printf("Enter Course ID to Search:");
    scanf("%d",&searchID);
    for(int i=0;i<total_courses;i++){
        if(details[i].courseID==searchID){
            printf("Enter details to update: ");
            printf("Enter Updated Course Name: ");
            scanf(" %[^\n]s",details[i].name);
            printf("Enter updated Duration of course: ");
            scanf("%f",&details[i].duration);
            printf("Enter new Credit score: ");
            scanf("%f",&details[i].credits);
```

```
        }
    }

}

void addCourse(struct Course *details, int *total_courses){
    printf("Add new Course:\n");
    printf("Enter Course ID: ");
    scanf("%d",&details[*total_courses].courseID);
    printf("Enter Course Name: ");
    scanf(" %[^\n]s",details[*total_courses].name);
    printf("Enter Duration of course: ");
    scanf("%f",&details[*total_courses].duration);
    printf("Enter Credits: ");
    scanf("%f",&details[*total_courses].credits);

    printf("Enter Instructor details:");
    printf("Enter Course type: ");
    scanf(" %[^\n]s",details[*total_courses].instructorDetails.course);
    printf("Enter name of Instructor:");
    scanf(" %[^\n]s",details[*total_courses].instructorDetails.instructor);

    (*total_courses)++;
    printf("\nCourse added Succefully!\n");
```

```
}

void viewCourse(struct Course *details, int total_courses){
    if(total_courses==0){
        printf("\nNo added Course\n");
        return;
    }
    printf("View Entire Course!\n");
    for(int i=0;i<total_courses;i++){
        printf("Course ID: %d\n",details[i].courseID);
        printf("Course Name: %s\n",details[i].name);
        printf("Duration: %0.2f\n",details[i].duration);
        printf("Credits: %0.2f\n",details[i].credits);
        printf("Course type: %s\n",details[i].instructorDetails.course);
        printf("Name of Instructor: %s\n",details[i].instructorDetails.instructor);

    }

}
```