CSE 115L: Programming Language I Lab (Section: 06)

Spring 2020

Lab-11 (Structures)

Structure is the collection of variables of different types under a single name for better handling.

Defining structure (syntax)	**Defining structure (example)**

struct [structure_tag]
{
   //member variable 1
   //member variable 2
   //member variable 3
   ...
}[structure_variables];
	struct Student
	{
	   char name[25];
	   int age;
	   char gender;
	};

Declaring structure variables separately	**Declaring structure variables with structure definition**

struct Student
{
   char name[25];
   int age;
   char gender;
};
 	
struct Student S1, S2;   //declaring variables of struct Student
	struct Student
	{
	   char name[25];
	   int age;
	   char gender;
	}S1, S2;

Structure initialization	**Accessing structure members**

struct Patient
{
   float height;
   int weight;
   int age;
};
	#include<stdio.h>
	
	struct Student
	{
	         char name[25];
	         int age;
	         char gender;
	};

struct Patient p1;
p1.height = 180.75;   //initialization of each member separately
p1.weight = 73;
p1.age = 23;
 	
//struct Patient p1 = { 180.75 , 73, 23 };
	int main(void)
	{
	struct Student s1;
	
	scanf(" %d ", &s1.age);
	gets(s1.name);
	
	printf("Name of Student 1: %s\n", s1.name);
	printf("Age of Student 1: %d\n", s1.age);
	
	return 0;
	}

Declaring Structure Variables Separately	**Declaring Structure Variables with Structure Definition**	**Example of structure definition using typedef**

#include <stdio.h>
#include <string.h>
 	
struct book_data
{
char title[100];
char author[100];
char topic[100];
int id;
};
 	
int main(void)
{
struct book_data b;
 	
strcpy(b.title, "Title");
strcpy(b.author, "Author");
strcpy(b.topic, "Topic");
b.id = 12;
 	
return 0;
}
	#include <stdio.h>
	#include <string.h>
	
	struct book_data
	{
	char title[100];
	char author[100];
	char topic[100];
	int id;
	}b;
	
	int main(void)
	{
	strcpy(b.title, "Title");
	strcpy(b.author, "Author");
	strcpy(b.topic, "Topic");
	b.id = 12;
	
	return 0;
	}
		#include <stdio.h>
		#include <string.h>
		
		typedef struct book_data
		{
		char title[100];
		char author[100];
		char topic[100];
		int id;
		}Book;
		
		//typedef struct book_data Book; //another way of declaring typedef
		
		int main(void)
		{
		//struct book_data b;
		Book b;
		strcpy(b.title, "Title");
		strcpy(b.author, "Author");
		strcpy(b.topic, "Topic");
		b.id = 12;
		return 0;}

Example: Nested Structure

```
#include <stdio.h>

struct Person
{
  char name[50];
  int age;
  struct Faculty
        {
        char designation[50];
        int courseNo;
        }f;
};

int main()
{
        struct Person p;

        printf("Enter name: ");
        gets(p.name);

        printf("Enter age: ");
        scanf("%d", &p.age);
        fflush(stdin);

        printf("Enter designation: ");
        gets(p.f.designation);

        printf("Enter total course taking: ");
        scanf("%d", &p.f.courseNo);

        printf("\n\nDisplaying:\n");
        printf("Name: %s\n", p.name);
        printf("Age: %d\n", p.age);
        printf("Designation: %s\n",
p.f.designation);

        return 0;
}
```

```
#include <stdio.h>

struct Faculty
{
        char designation[50];
        int courseNo;
};
struct Person
{
  char name[50];
  int age;
  struct Faculty f;
};

int main()
{
        struct Person p;

        printf("Enter name: ");
        gets(p.name);

        printf("Enter age: ");
        scanf("%d", &p.age);
        fflush(stdin);

        printf("Enter designation: ");
        gets(p.f.designation);

        printf("Enter total course taking: ");
        scanf("%d", &p.f.courseNo);

        printf("\n\nDisplaying:\n");
        printf("Name: %s\n", p.name);
        printf("Age: %d\n", p.age);
        printf("Designation: %s\n",
p.f.designation);

        return 0;
}
```

Example: Add two distances

```
#include <stdio.h>

struct Distance
{
        int feet;
        float inch;
};

int main(void)
{
        struct Distance dist1, dist2, sum;

        printf("1st distance\n");
        printf("Enter feet: ");
        scanf("%d", &dist1.feet);

        printf("Enter inch: ");
        scanf("%f", &dist1.inch);
        printf("2nd distance\n");

        printf("Enter feet: ");
        scanf("%d", &dist2.feet);

        printf("Enter inch: ");
        scanf("%f", &dist2.inch);

        // adding feet
        sum.feet = dist1.feet + dist2.feet;
        // adding inches
        sum.inch = dist1.inch + dist2.inch;

        // changing to feet if inch is greater than 12
        while (sum.inch >= 12)
        {
        ++sum.feet;
        sum.inch = sum.inch - 12;
        }

        printf("Sum of distances = %d\'-%.1f\"", sum.feet, sum.inch);
        return 0;
}
```

Example: Array of structures

```
#include<stdio.h>

typedef struct person
{
        char name[50];
        int id;
}student;


int main()
{
        int i;
        student st[2];

        for(i=0; i<2; i++)
        {
        printf("Enter student# %d name and id:\n",i+1);
        gets(st[i].name);
        scanf("%d",&st[i].id);
        fflush(stdin);
        }

        for(i=0; i<2; i++)
        {
        printf("\n\nPrint student# %d name and id:\n",i+1);
        printf("Name: %s\n", st[i].name);
        printf("ID: %d\n", st[i].id);
        }
        return 0;
}
```

Structures and Pointers:

Create pointers to struct	Access members using pointers

struct name {
  // member1;
  // member2;
   .
   .
};
 	
int main()
{
   struct name *ptr, Harry;
}
	#include <stdio.h>
	
	struct person
	{
	  int age;
	  float weight;
	};
	
	int main()
	{
	    struct person *personPtr, person1, person2;
	    personPtr = &person1;
	
	    printf("Enter age: ");
	    scanf("%d", **&personPtr->age**);
	
	    printf("Enter weight: ");
	    scanf("%f", **&personPtr->weight**);
	
	    printf("Displaying:\n");
	    printf("Age: %d\n", **(*personPtr).age**);
	    printf("weight: %.2f", **personPtr->weight**);
	
	    return 0;
	}

Structures and Function:

Structure as function parameter	**Function returns Structure**

#include<stdio.h>
 	
struct Student
{
char name[10];
int id;
};
 	
void show(struct Student st); //prototype
 	
int main(void)
{
struct Student s;
printf("\nEnter Student record:\n");
printf("Student name:\t");
scanf("%s", s.name);
printf("\nEnter Student ID no.:\t");
scanf("%d", &s.id);
show(s);
return 0;
}
 	
void show(struct Student st)
{
printf("\nstudent name is %s", st.name);
printf("\nID is %d", st.id);
}
	#include <stdio.h>
	
	// creating a student structure template
	struct Student {
	char name[100];
	int id;
	};
	
	// function declaration
	struct Student getDetail(void);
	void show(struct Student st);
	
	int main(void)
	{
	// creating a student structure array variable
	struct Student stdArr[3];
	// other variables
	int i;
	// taking user input
	for (i = 0; i < 3; i++) {
	printf("Enter detail of student #%d\n", (i+1));
	stdArr[i] = getDetail();
	fflush(stdin);
	}
	
	// output
	for (i = 0; i < 3; i++) {
	printf("\nStudent #%d Detail:", (i+1));
	show(stdArr[i]);
	printf("\n");
	}
	
	return 0;
	}
	
	struct Student getDetail(void) {
	// temp structure variable
	struct Student std;
	
	printf("Enter Name: ");
	gets(std.name);

```
  printf("Enter ID: ");
  scanf("%d", &std.id);

  return std;
}

void show(struct Student st)
{
	printf("\nstudent name is %s", st.name);
	printf("\nID is %d", st.id);
}
```