Programming and Data Structures - I

Lecture 12

Kripabandhu Ghosh

CDS, IISER Kolkata

STRUCTURES

Structures

Format

```
struct tag_name
{
   data_type member1;
   data_type member2;
   ...
}
```

- collection of data items of different data types under the same name **tag_name**
- provision of having logically connected variables under the same name

Example

```
struct student
{
   char name[25];
   char roll_no[10];
   int semester_no;
};
```

Structures: declaration

```
struct student
{
   char name[25];
   char roll_no[10];
   int semester_no;
};
struct student student1, student2;
```

```
struct student
{
   char name[25];
   char roll_no[10];
   int semester_no;
} student1, student2;
```

```
struct
{
   char name[25];
   char roll_no[10];
   int semester_no;
} student1, student2;
```

Structures: member value assignment

Assignment

strcpy(student1.name, “Kripa”);
strcpy(student1.roll_no, “F0172”);
student1.semester_no = 5;

Initialization

student2 = {“Kripa”, “F0172”, 5};

Copy

struct student student3;
student3 = student1;

Structures: example

Program

```
#include<stdio.h>
#include<string.h>

struct student
{
   char name[25];
   char roll_no[10];
   int semester_no;
} student1;
```

Structures: example

Program

```
int main(void)
{
   struct student student2, student3 = {"Alok", "17MS345", 4}, student4;
   strcpy(student1.name, "Kripa");
   strcpy(student1.roll_no, "F0172");
   student1.semester_no = 5;
   student4 = student1;//Copy
   printf("Please input name, roll no and semester no below:\n");
   scanf("%s %s %d", student2.name, student2.roll_no, &student2.semester_no);
   printf("Student 1 record: %s %s %d\n", student1.name, student1.roll_no,
student1.semester_no);
   printf("Student 2 record: %s %s %d\n", student2.name, student2.roll_no,
student2.semester_no);
   printf("Student 3 record: %s %s %d\n", student3.name, student3.roll_no,
student3.semester_no);
   printf("Student 4 record: %s %s %d\n", student4.name, student4.roll_no,
student4.semester_no);
   printf("%lu %lu\n", sizeof(student1), sizeof(student2));
   return 0; }
```

Structures: example

Output

Please input name, roll no and semester no below:
Anuj
18MS123
5
Student 1 record: Kripa F0172 5
Student 2 record: Anuj 18MS123 5
Student 3 record: Alok 17MS345 4
Student 4 record: Kripa F0172 5
40 40

Structures: functions

Program

```
#include<stdio.h>
#include<string.h>

typedef struct book{
   char title[50];
   char id[10];
   int no;
}BOOK;

void display(BOOK b)
{
   printf("Book title: %s\n", b.title);
   printf("Book id: %s\n", b.id);
   printf("Copies in stock: %d\n\n", b.no);
}

BOOK update(BOOK b)
{
   b.no++;
   return b;
}
```

Structures: functions (contd.)

Program

```
int main(void)
{
   BOOK book1 = {"Let us C", "B123", 5}, book2 = {"The Art of Computer
Programming", "A1245", 3};
   display(book1);
   display(book2);
   book1 = update(book1);
   display(book1);
   return 0;
}
```

Structures: functions (contd.)

Output

```
Book title: Let us C
Book id: B123
Copies in stock: 5

Book title: The Art of Computer Programming
Book id: A1245
Copies in stock: 3

Book title: Let us C
Book id: B123
Copies in stock: 6
```

Pointer to structures

Program

```
#include<stdio.h>
#include<string.h>

typedef struct book
{
  char title[50];
  char id[10];
  int no;
}BOOK;

void display(BOOK* b)
{
  printf("Book title: %s\n", b->title);
  printf("Book id: %s\n", b->id);
  printf("Copies in stock: %d\n\n", b->no);
}

void update(BOOK* b)
{
  b->no++;
}

int main(void)
{
  BOOK book1 = {"Let us C", "B123", 5}, book2 = {"The Art of Computer Programming", "A1245", 3};
  display(&book1);
  display(&book2);
  update(&book1);
  display(&book1);
  return 0;
}
```

Pointer to structures (contd.)

Output

```
Book title: Let us C
Book id: B123
Copies in stock: 5

Book title: The Art of Computer Programming
Book id: A1245
Copies in stock: 3

Book title: Let us C
Book id: B123
Copies in stock: 6
```


Thank you.