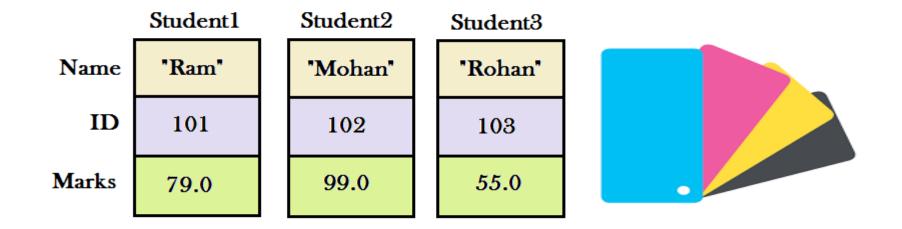
## Structure in C & Miscellaneous



## Passing One-dimensional Array to a Function

```
#include <stdio.h>
void display(int age)
  printf("%d", age);
int main()
  int ageArray[] = \{2, 3, 4\};
  display(ageArray[2]);
  return 0;
```

Output: 4

# Passing One-dimensional Array to a Function

```
#include <stdio.h>
float average(float age[]);
int main()
 float avg, age[] = \{23.4, 55, 22.6, 3,
40.5, 18};
 avg = average(age);
 printf("Average age = %.2f", avg);
 return 0;
```

```
float average(float age[])
       int i;
       float avg, sum = 0.0;
       for (i = 0; i < 6; ++i) {
               sum += age[i];
       avg = (sum / 6);
       return avg;
```

Output: Average age = 27.08

# Passing Two-dimensional Array to a Function

```
#include <stdio.h>
void displayNumbers(int num[2][2]);
int main()
  int num[2][2], i, j;
  printf("Enter 4 numbers:\n");
  for (i = 0; i < 2; ++i)
    for (j = 0; j < 2; ++j)
       scanf("%d", &num[i][j]);
  displayNumbers(num);
  return 0;
```

```
\label{eq:coid_displayNumbers} \begin{tabular}{ll} void_displayNumbers(int_n[2][2]) \\ \{ & int_i, j; \\ printf("Displaying:\n"); \\ for_(i=0; i<2; ++i) \\ for_(j=0; j<2; ++j) \\ printf("\%d", n[i][j]); \\ \} \end{tabular}
```

#### **Output:**

Enter 4 numbers: 2 3 4 5

Displaying: 2 3 4 5

### Passing Array to a Function & Sort it

```
#include <stdio.h>
void bubble_sort(long [], long);
int main()
 int array[100], n, c, d, swap;
 array = \{10, 1, 2, 6\};
 bubble_sort(array, n);
 printf("Sorted list in ascending order:\n");
 for (c = 0; c < n; c++)
   printf("%ld,", array[c]);
```

```
void bubble_sort(long list[], long n){
 long c, d, t;
 for (c = 0; c < (n - 1); c++)
  for (d = 0; d < n - c - 1; d++)
    if (\operatorname{list}[d] > \operatorname{list}[d+1])
     t = list[d];
     list[d] = list[d+1];
     list[d+1] = t;
```

Output: 1,2,6,10

# Syntax of structure

```
struct structure_name
{
    data_type member1;
    data_type member2;
    .
    data_type memeber;
};
```

```
struct Person
                          Declare
  char name[50];
                          after
  int citNo;
                          header
  float salary;
                          file
int main()
  struct Person person1, person2, p[20];
  return 0;
```

# Syntax of structure

```
Another way structure declaration:
struct Person
{
    char name[50];
    int citNo;
    float salary;
} person1, person2, p[20];
```

#### How to Access members of a structure?

- Member operator(.)
- Structure pointer operator(->)
- Person1.salary, p[5].name etc

## Structure Example

```
#include <stdio.h>
#include <string.h>
struct student
{
    int id;
    char name[20];
    float percentage;
};
```

Id is: 1

Name is: Raju

Percentage is: 86.50

```
int main()
   struct student record;
   record.id=1;
   strcpy(record.name, "Raju");
   record.percentage = 86.5;
   printf(" Id is: %d \n", record.id);
   printf(" Name is: %s \n", record.name);
   printf(" Percentage is: %.2f \n",
record.percentage);
   return 0;
```

## Structure Example

```
#include <stdio.h>
struct Distance
  int feet;
  float inch;
} dist1, dist2, sum;
int main(){
  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 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;
```

## Structure Example

1st distance

Enter feet: 12

Enter inch: 7.9

2nd distance

Enter feet: 2

Enter inch: 9.8

Sum of distances = 15'-5.7"

# Keyword typedef

```
struct distance{
   int feet;
   float inch;
};

int main() {
   structure distance d1, d2;
}
```

```
typedef struct distance{
  int feet;
  float inch;
} DISTANCE;

int main() {
  DISTANCE dist1, dist2, sum;
}
```