Assignment (11/8/2025)

## **Q1 – Define a structure for student record and print details**

**IPO**  
**Input:** Name, Roll No, Marks  
**Process:** Store in structure and display  
**Output:** Student details

#include <stdio.h>

struct Student {

char name[50];

int roll;

float marks;

};

int main() {

struct Student s;

printf("Enter name: ");

scanf("%s", s.name);

printf("Enter roll no: ");

scanf("%d", &s.roll);

printf("Enter marks: ");

scanf("%f", &s.marks);

printf("\n--- Student Details ---\n");

printf("Name : %s\nRoll : %d\nMarks: %.2f\n", s.name, s.roll, s.marks);

return 0;

}

**Sample Output**

Enter name: Arun

Enter roll no: 101

Enter marks: 87.5

--- Student Details ---

Name : Arun

Roll : 101

Marks: 87.50

## **Q2 – Store and display employee details using structures**

**IPO**  
**Input:** Name, ID, Salary  
**Process:** Store in structure and display  
**Output:** Employee details

#include <stdio.h>

struct Employee {

char name[50];

int id;

float salary;

};

int main() {

struct Employee e;

printf("Enter name: ");

scanf("%s", e.name);

printf("Enter ID: ");

scanf("%d", &e.id);

printf("Enter salary: ");

scanf("%f", &e.salary);

printf("\n--- Employee Details ---\n");

printf("Name : %s\nID : %d\nSalary: %.2f\n", e.name, e.id, e.salary);

return 0;

}

**Sample Output**

Enter name: Ravi

Enter ID: 2001

Enter salary: 45000

--- Employee Details ---

Name : Ravi

ID : 2001

Salary: 45000.00

## **Q3 – Pass a structure to a function**

**IPO**  
**Input:** Name, Roll, Marks  
**Process:** Pass structure to function to print details  
**Output:** Student details

#include <stdio.h>

struct Student {

char name[50];

int roll;

float marks;

};

void display(struct Student s) {

printf("\nName : %s\nRoll : %d\nMarks: %.2f\n", s.name, s.roll, s.marks);

}

int main() {

struct Student s;

printf("Enter name: ");

scanf("%s", s.name);

printf("Enter roll: ");

scanf("%d", &s.roll);

printf("Enter marks: ");

scanf("%f", &s.marks);

display(s);

return 0;

}

**Sample Output**

Enter name: Meena

Enter roll: 102

Enter marks: 91.5

Name : Meena

Roll : 102

Marks: 91.50

## **Q4 – Store multiple student records using array of structures**

**IPO**  
**Input:** Names, Rolls, Marks of n students  
**Process:** Store in array of structures and display  
**Output:** All student records

#include <stdio.h>

struct Student {

char name[50];

int roll;

float marks;

};

int main() {

struct Student s[3];

for(int i=0;i<3;i++) {

printf("Enter name, roll, marks of student %d: ", i+1);

scanf("%s %d %f", s[i].name, &s[i].roll, &s[i].marks);

}

printf("\n--- Student Records ---\n");

for(int i=0;i<3;i++) {

printf("%s\t%d\t%.2f\n", s[i].name, s[i].roll, s[i].marks);

}

return 0;

}

**Sample Output**

Enter name, roll, marks of student 1: Arun 101 87.5

Enter name, roll, marks of student 2: Meena 102 91.5

Enter name, roll, marks of student 3: Kiran 103 85.0

--- Student Records ---

Arun 101 87.50

Meena 102 91.50

Kiran 103 85.00

## **Q5 – Demonstrate nested structures**

**IPO**  
**Input:** Student name, roll, date of birth (day, month, year)  
**Process:** Use one structure inside another  
**Output:** Student details with DOB

#include <stdio.h>

struct Date {

int day, month, year;

};

struct Student {

char name[50];

int roll;

struct Date dob;

};

int main() {

struct Student s;

printf("Enter name: ");

scanf("%s", s.name);

printf("Enter roll: ");

scanf("%d", &s.roll);

printf("Enter DOB (dd mm yyyy): ");

scanf("%d %d %d", &s.dob.day, &s.dob.month, &s.dob.year);

printf("\n--- Student Details ---\n");

printf("Name : %s\nRoll : %d\nDOB : %02d-%02d-%04d\n",

s.name, s.roll, s.dob.day, s.dob.month, s.dob.year);

return 0;

}

**Sample Output**

Enter name: Arjun

Enter roll: 105

Enter DOB (dd mm yyyy): 15 08 2005

--- Student Details ---

Name : Arjun

Roll : 105

DOB : 15-08-2005

## **Q6 – Calculate total and average marks using structures**

**IPO**  
**Input:** Marks in 3 subjects  
**Process:** Sum and divide by 3  
**Output:** Total and average

#include <stdio.h>

struct Marks {

float m1, m2, m3, total, avg;

};

int main() {

struct Marks s;

printf("Enter marks in 3 subjects: ");

scanf("%f %f %f", &s.m1, &s.m2, &s.m3);

s.total = s.m1 + s.m2 + s.m3;

s.avg = s.total / 3;

printf("Total: %.2f\nAverage: %.2f\n", s.total, s.avg);

return 0;

}

**Sample Output**

Enter marks in 3 subjects: 80 90 85

Total: 255.00

Average: 85.00

## **Q7 – Find the highest marks among students**

**IPO**  
**Input:** Marks of n students  
**Process:** Compare and find highest  
**Output:** Highest marks and student name

#include <stdio.h>

struct Student {

char name[50];

float marks;

};

int main() {

struct Student s[3];

int i, pos=0;

for(i=0;i<3;i++) {

printf("Enter name and marks of student %d: ", i+1);

scanf("%s %f", s[i].name, &s[i].marks);

if(s[i].marks > s[pos].marks)

pos = i;

}

printf("\nHighest Marks: %.2f by %s\n", s[pos].marks, s[pos].name);

return 0;

}

**Sample Output**

Enter name and marks of student 1: Arun 87

Enter name and marks of student 2: Meena 92

Enter name and marks of student 3: Kiran 89

Highest Marks: 92.00 by Meena

## **Q8 – Sort student records by name**

**IPO**  
**Input:** Name, Roll of n students  
**Process:** Sort alphabetically by name  
**Output:** Sorted list

#include <stdio.h>

#include <string.h>

struct Student {

char name[50];

int roll;

};

int main() {

struct Student s[3], temp;

for(int i=0;i<3;i++) {

printf("Enter name and roll: ");

scanf("%s %d", s[i].name, &s[i].roll);

}

for(int i=0;i<3;i++) {

for(int j=i+1;j<3;j++) {

if(strcmp(s[i].name, s[j].name) > 0) {

temp = s[i];

s[i] = s[j];

s[j] = temp;

}

}

}

printf("\n--- Sorted Records ---\n");

for(int i=0;i<3;i++)

printf("%s\t%d\n", s[i].name, s[i].roll);

return 0;

}

**Sample Output**

Enter name and roll: Kiran 103

Enter name and roll: Arun 101

Enter name and roll: Meena 102

--- Sorted Records ---

Arun 101

Kiran 103

Meena 102

## **Q9 – Using union to store data of different types**

**IPO**  
**Input:** Roll (int), Marks (float), Grade (char)  
**Process:** Store in union (only one value at a time)  
**Output:** Display value stored last

#include <stdio.h>

union Data {

int roll;

float marks;

char grade;

};

int main() {

union Data d;

d.roll = 101;

printf("Roll : %d\n", d.roll);

d.marks = 87.5;

printf("Marks: %.2f\n", d.marks);

d.grade = 'A';

printf("Grade: %c\n", d.grade);

return 0;

}

**Sample Output**

Roll : 101

Marks: 87.50

Grade: A

(Note: Only last assigned value is valid in union.)

## **Q10 – Compare structure vs union with sample program**

**IPO**  
**Input:** Assign different values in structure and union  
**Process:** Show how structure stores all, union stores one  
**Output:** Memory behavior and output difference

#include <stdio.h>

#include <string.h>

struct Student {

int roll;

float marks;

char grade;

};

union StudentU {

int roll;

float marks;

char grade;

};

int main() {

struct Student s = {101, 87.5, 'A'};

union StudentU u;

u.roll = 101;

u.marks = 87.5;

u.grade = 'A';

printf("Structure: %d %.2f %c\n", s.roll, s.marks, s.grade);

printf("Union : %d %.2f %c\n", u.roll, u.marks, u.grade);

return 0;

}

**Sample Output**

Structure: 101 87.50 A

Union : 1092616192 87.50 A