Day: Structures and Unions – 11/08/2025

1. Define a Structure for Student Record and Print Details

Code:

```
#include <stdio.h>
struct Student {char name[50]; int roll; float marks;};
int main(){struct Student s={"John Doe",101,85.5};
printf("Name: %s\n",s.name);printf("Roll No: %d\n",s.roll);
printf("Marks: %.2f\n",s.marks);return 0;}
```

Output Screenshot:

2. Store and Display Employee Details Using Structures

Code:

```
#include <stdio.h>
struct Employee {char name[50]; int id; float salary;};
int main(){struct Employee e={"Alice",2001,55000.75};
printf("Name: %s\n",e.name);printf("ID: %d\n",e.id);
printf("Salary: %.2f\n",e.salary);return 0;}
```

Output Screenshot:

3. Pass a Structure to a Function

Code:

```
#include <stdio.h>
struct Point {int x,y;};
void display(struct Point p){printf("X: %d, Y: %d\n",p.x,p.y);}
int main(){struct Point pt={10,20};display(pt);return 0;}
```

Output Screenshot:

4. Store Multiple Student Records Using Array of Structures

Code:

```
#include <stdio.h>
struct Student {char name[50]; int roll; float marks;};
int main(){struct Student
s[3]={{"John",1,80.5},{"Alice",2,85.0},{"Bob",3,78.0}};
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;}
```

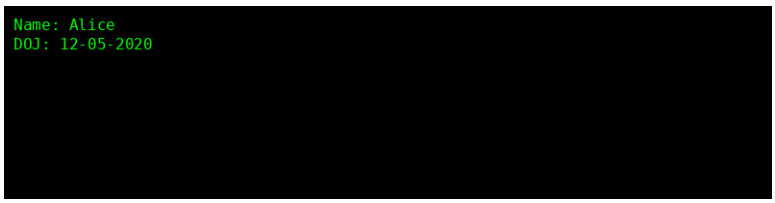
Output Screenshot:

5. Demonstrate Nested Structures

Code:

```
#include <stdio.h>
struct Date {int day,month,year;};
struct Employee {char name[50]; struct Date doj;};
int main(){struct Employee e={"Alice",{12,5,2020}};
printf("Name: %s\n",e.name);printf("DOJ: %02d-%02d-
%d\n",e.doj.day,e.doj.month,e.doj.year);return 0;}
```

Output Screenshot:

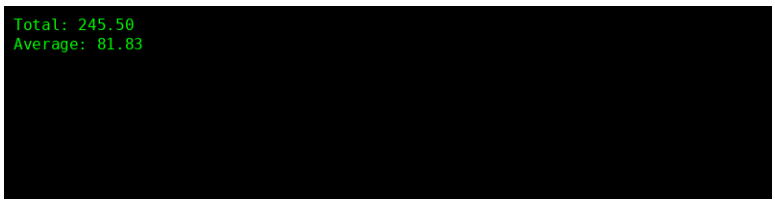


6. Calculate Total and Average Marks Using Structures

Code:

```
#include <stdio.h>
struct Student {char name[50]; float marks[3];};
int main(){struct Student s={"John",{80.5,75.0,90.0}};float total=0;
for(int i=0;i<3;i++)total+=s.marks[i];printf("Total: %.2f\n",total);
printf("Average: %.2f\n",total/3);return 0;}
```

Output Screenshot:

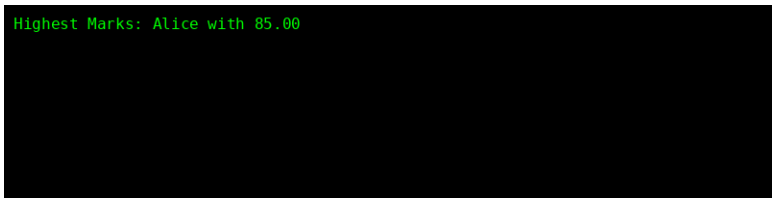


7. Find Highest Marks Among Students

Code:

```
#include <stdio.h>
struct Student {char name[50]; float marks;};
int main(){struct Student
s[3]={{"John",80.5},{"Alice",85.0},{"Bob",78.0}};
int highestIndex=0;for(int
i=1;i<3;i++){if(s[i].marks>s[highestIndex].marks)highestIndex=i;}
printf("Highest Marks: %s with
%.2f\n",s[highestIndex].name,s[highestIndex].marks);return 0;}
```

Output Screenshot:



8. Sort Student Records by Name Using Structure

Code:

```
#include <stdio.h>
#include <string.h>
struct Student {char name[50]; int roll;};
int main(){struct Student
```

```
s[3]={{"John",1},{"Alice",2},{"Bob",3}};struct Student temp;
for(int i=0;i<2;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;}}}
for(int i=0;i<3;i++)printf("%s\t%d\n",s[i].name,s[i].roll);return 0;}
```

Output Screenshot:

9. Union to Store Data of Different Types

Code:

```
#include <stdio.h>
#include <string.h>
union Data {int i; float f; char str[20];};
int main(){union Data data;data.i=10;printf("Integer:
%d\n",data.i);data.f=220.5;
printf("Float: %.2f\n",data.f);strcpy(data.str,"Hello");printf("String:
%s\n",data.str);return 0;}
```

Output Screenshot:

10. Compare and Contrast Structure vs Union

Code:

```
#include <stdio.h>
#include <string.h>
struct MyStruct {int i; float f; char str[20];};
union MyUnion {int i; float f; char str[20];};
int main(){struct MyStruct s={10,220.5,"Hello"};union MyUnion
u;u.i=10;
printf("Structure: %d, %.2f,
```

%s\n",s.i,s.f,s.str);u.f=220.5;strcpy(u.str,"Hello");
printf("Union: %d, %.2f, %s\n",u.i,u.f,u.str);return 0;}

Output Screenshot: