Content 13

Structures, Unions & Enums in C++

**Topic Involved;**

* Structure.
* Typedef.
* Union.
* Enum.

#### Structures in C++

The structure is a user-defined data type that is available in C++. Structures are used to combine different types of data types, just like an array is used to combine the same type of data types.

// Explaination of structures.

#include <iostream>

using namespace std;

struct student

{

    char sNamecharachter;

    int sId;

    float marks;

} s1, s2;

int main()

{

    s1.sNamecharachter='A';

    s1.sId=01;

    s1.marks=45.2;

    cout<<"The starting Character of student 1: "<<s1.sNamecharachter<<endl;

    cout<<"The Id of student 1: "<<s1.sId<<endl;

    cout<<"The marks of student 1: "<<s1.marks<<endl;

    s2.sNamecharachter='N';

    s2.sId=02;

    s2.marks=50.6;

    cout<<"\n\nThe starting Character of student 2: "<<s2.sNamecharachter<<endl;

    cout<<"The Id of student 2: "<<s2.sId<<endl;

    cout<<"The marks of student 2: "<<s2.marks<<endl;

    return 0;

}

**Output:**

The starting Character of student 1: A

The Id of student 1: 1

The marks of student 1: 45.2

The starting Character of student 2: N

The Id of student 2: 2

The marks of student 2: 50.6

**Typedef;**

we have used a keyword “**typedef**” before struct and after the closing bracket of structure, we have written “ep”. Now we can create structure variables without using the keyword “struct” and name of the struct.

**Code For Explaination:**

// Explaination of typedef.

#include <iostream>

using namespace std;

typedef struct student

{

    char sNamecharachter;

    int sId;

    float marks;

} stu;

int main()

{   stu s1,s2;

    s1.sNamecharachter='A';

    s1.sId=01;

    s1.marks=45.2;

    cout<<"The starting Character of student 1: "<<s1.sNamecharachter<<endl;

    cout<<"The Id of student 1: "<<s1.sId<<endl;

    cout<<"The marks of student 1: "<<s1.marks<<endl;

    return 0;

}

**Output:**

The starting Character of student 1: A

The Id of student 1: 1

The marks of student 1: 45.2

#### Unions in C++

Unions are similar to structures but they provide better memory management then structures.  Unions use shared memory so only 1 variable can be used at a time.

**Code for unions;**

// For Unions

#include <iostream>

using namespace std;

union student

{

    char sNamecharachter;

    int sId;

    float marks;

} s1,s2;

int main()

{   union student s1;

    s1.sNamecharachter='A';

    cout<<"The starting Character of student 1: "<<s1.sNamecharachter<<endl;

    return 0;

}

**Output:**

The starting Character of student 1: A

**Important points about Unions.**

* We can only use 1 variable at a time otherwise the compiler will give us a garbage value
* The compiler chooses the data type which has maximum memory for the allocation.

#### Enums in C++

Enums are user-defined types which consist of named constants. Enums are used to make the program more readable.

**Code for Enum:**

// For Enum

#include<iostream>

using namespace std;

int main()

{

enum meal{breakfast,launch=5,dinner};

meal m1=breakfast,m2=launch,m3=dinner;

cout<<m1<<endl;

cout<<m2<<endl;

cout<<m3<<endl;

cout<<"\n\n"<<(m1==2)<<endl; //if it is true then it will return 1 otherwise ) as false

return 0;

}

**Output:**

0

5

45

0