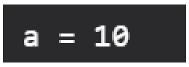
Q1. Write a C++ Program to shown the concept of Virtual Base Class also define a specific problem where we can use this concept for solving the specified problem using the Virtual Base Class concept.

```
#include <iostream>
using namespace std;

class A {
public:
    int a;
    A() // constructor
    {
        a = 10;
    }
};

class B : public virtual A {
};

class C : public virtual A {
};

class D : public B, public C {
};

int main()
{
    D object; // object creation of class d
    cout << "a = " << object.a << endl;

    return 0;
}
```



```
a = 10
```

Q2. Write a C++ Program to a base class student which contains the attributes like student roll number, name etc. and the operation included the information from the roll number, display the student information and assigned the value to the various attribute of the class. Now, this base class derived to new classes namely test and sports. The derived lasses also contain some distinct attributes related to the uniqueness of that class. Once again create a child class result by inheriting the features of test and sports class. Now, Write the definition of the child class result by removing the duplication of the inherited member due to the multiple paths from student class to results class.

```
#include<iostream>
using namespace std;
class student //base class declaration
{
protected:
int r_no;
public:
void get _n (int a)
{
r_no =a;
}
void put_n (void)
{
cout << “Roll No. : “<< r_no;
cout << “In”;
}
};
class test : public student
{ //Intermediate base class
protected : int parti, par 2;
public :
void get_m (int x, int y) {
parti = x; part 2 = y; }
void put_m (void) {
cout << “marks obtained: “ << “In”
<< “Part 1 = “ << part1 << “in”
<< “Part 2 = “ << part2 << “In”;
}
};
class sports // base for result
{
protected : int score;
public:
void get_s (int s) {
score = s }
void put_s (void) {
cout << “ sports wt. : “ << score << “\n\n”;
}
};
class result : public test, public sports //Derived from test
& sports
{
int total;
public:
void display (void);
};
void result : : display (void)
{
total = part1 + part2 + score;
put_n ( ) ;.
put_m ( );
put_S ( );
cout << “Total score: “ <<total<< “\n”
}
```

```
main ( )

{

clrscr ( ) ;

result S1;

S1.get_n (347) ;

S1.get_m (30, 35);

S1.get_s (7) ;

S1.display ( ) ;

}
```