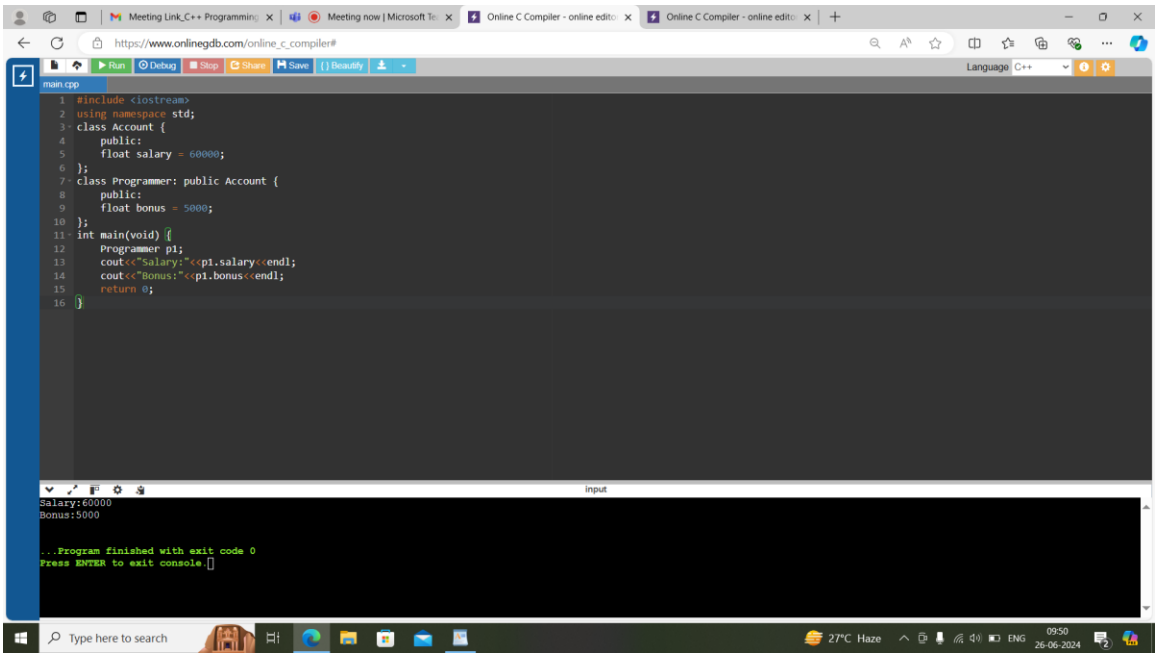
SINGLE INHERITENCE :-

```
#include <iostream>
using namespace std;
class Account {
	public:
	float salary = 60000;
};
class Programmer: public Account {
	public:
	float bonus = 5000;
};
int main(void) {
	Programmer p1;
	cout<<"Salary:"<<p1.salary<<endl;
	cout<<"Bonus:"<<p1.bonus<<endl;
	return 0;
}
```

OUTPUT :-

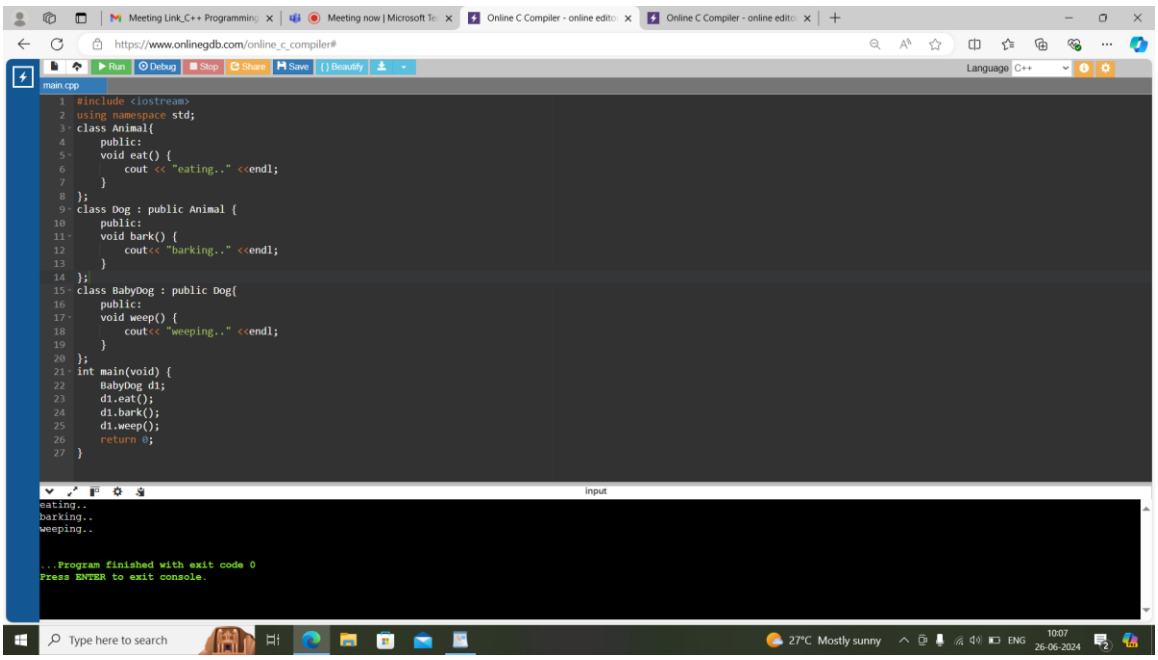


MULTILEVEL INHERITANCE :-

```
#include <iostream>
using namespace std;
class Animal{
    public:
    void eat() {
        cout << "eating.." <<endl;
    }
};
class Dog : public Animal {
    public:
    void bark() {
        cout<< "barking.." <<endl;
    }
};
class BabyDog : public Dog{
```

```
    public:
    void weep() {
        cout<< "weeping.." <<endl;
    }
};
int main(void) {
    BabyDog d1;
    d1.eat();
    d1.bark();
    d1.weep();
    return 0;
}
```

output :-



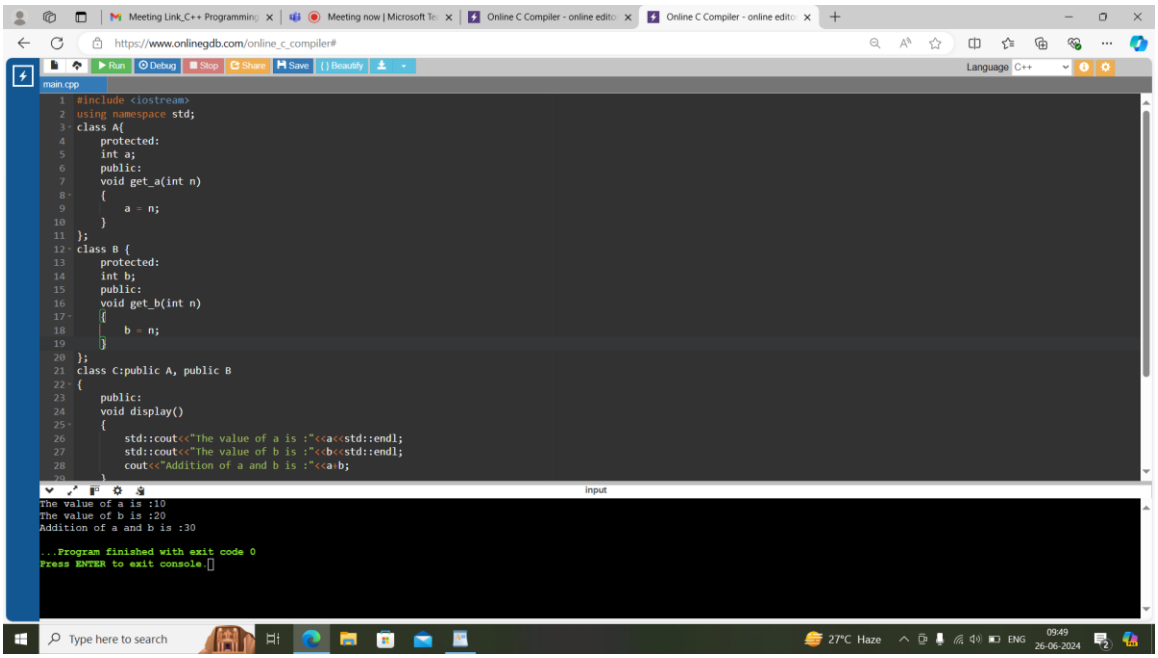
MULTIPLE INHERITANCE :-

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

```
class A{
	protected:
	int a;
	public:
	void get_a(int n)
	{
		a = n;
	}
};
class B {
	protected:
	int b;
	public:
	void get_b(int n)
	{
		b = n;
	}
};
class C:public A, public B
{
	public:
	void display()
	{
		std::cout<<"The value of a is :"<<a<<std::endl;
		std::cout<<"The value of b is :"<<b<<std::endl;
		cout<<"Addition of a and b is :"<<a+b;
	}
};
```

```
int main()
{
	C c;
	c.get_a(10);
	c.get_b(20);
	c.display();
	return 0;
}
```

OUTPUT :-



HYBRID INHERITANCE :-

```
#include <iostream>
using namespace std;
class vehicle
{
```

```
public:
vehicle()
	{
cout<< "This is a vehicle\n";
	}
};
class Car: public vehicle
{
public:
Car()
	{
cout<< "This is a car\n";
	}
};
class Racing
{
public:
Racing()
	{
cout<< "This is for Racing\n";
	}
};
class Ferrari: public Car, public Racing
{
public:
Ferrari()
	{
cout<< "Ferrari is a Racing Car\n";
```

```
    }


};
int main() {
    Ferrari f;
    return 0;
}
```

OUTPUT :-