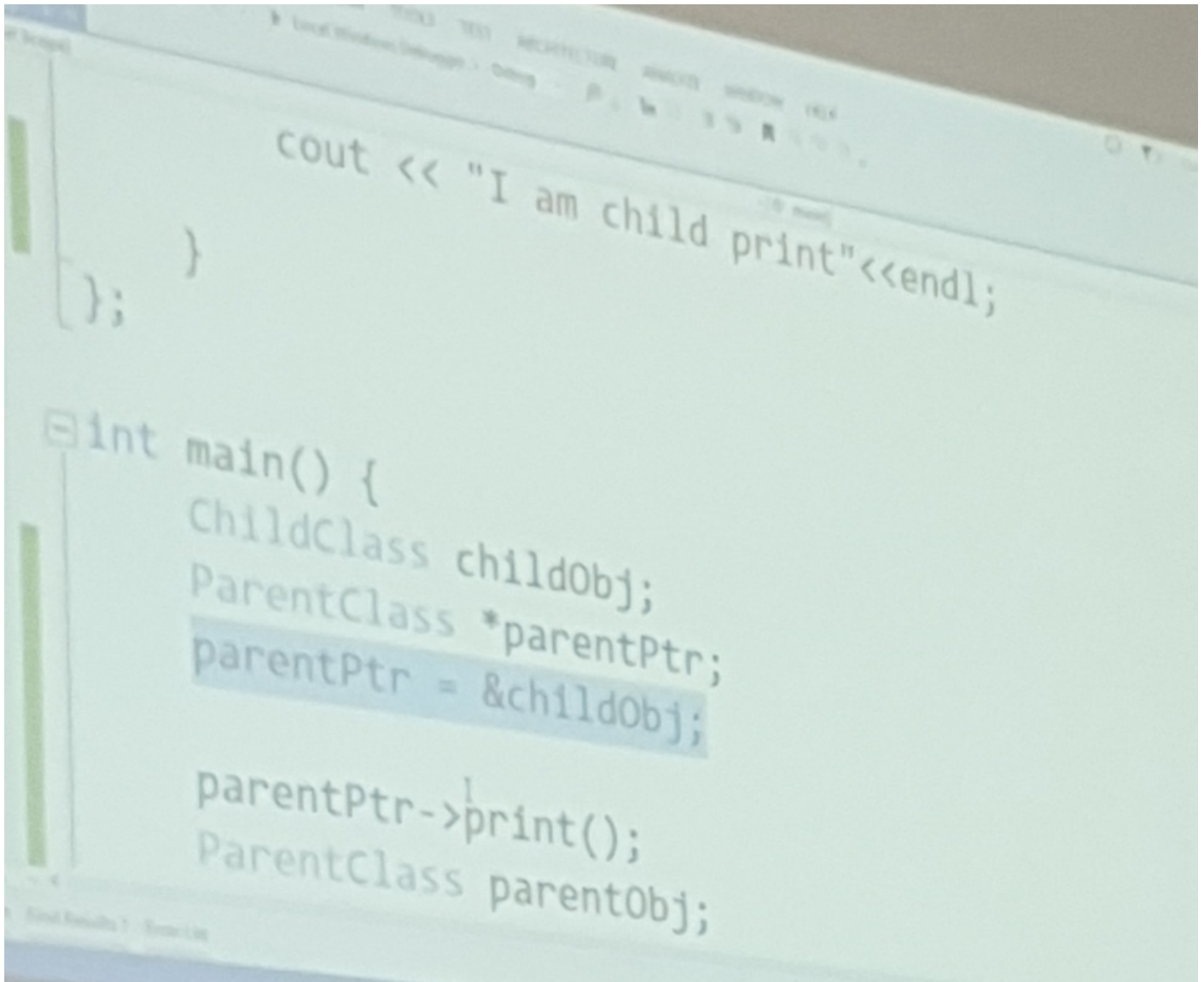


## OOP(Function overriding + Virtual Function)

### Function overriding



### Virtual Function

whenever, you have scenario, where you want to call the child class function using the parent class pointer (where the pointer holds the address of the child class object), use virtual keyword.

if you dont use virtual keyword, the parent pointer will call the parent class function even if holds the address the child class.

**Pure Virtual Function...**

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

class ParentClass{
public:
    virtual void print()=0{
        cout << "I am parent print"<<endl;
    }
};

class ChildClass : public ParentClass{
public:
    1
```

```
class Shape{
public:
    virtual void draw() = 0;
};

class circle : public Shape{
public:
    1

    void draw(){ // this is called function overriding.
        cout << "I am child print"<<endl;
    }
}
```

**Abstract class :** The class from which we can not make any instance / object.

**Concrete class :** From which we can create objects.