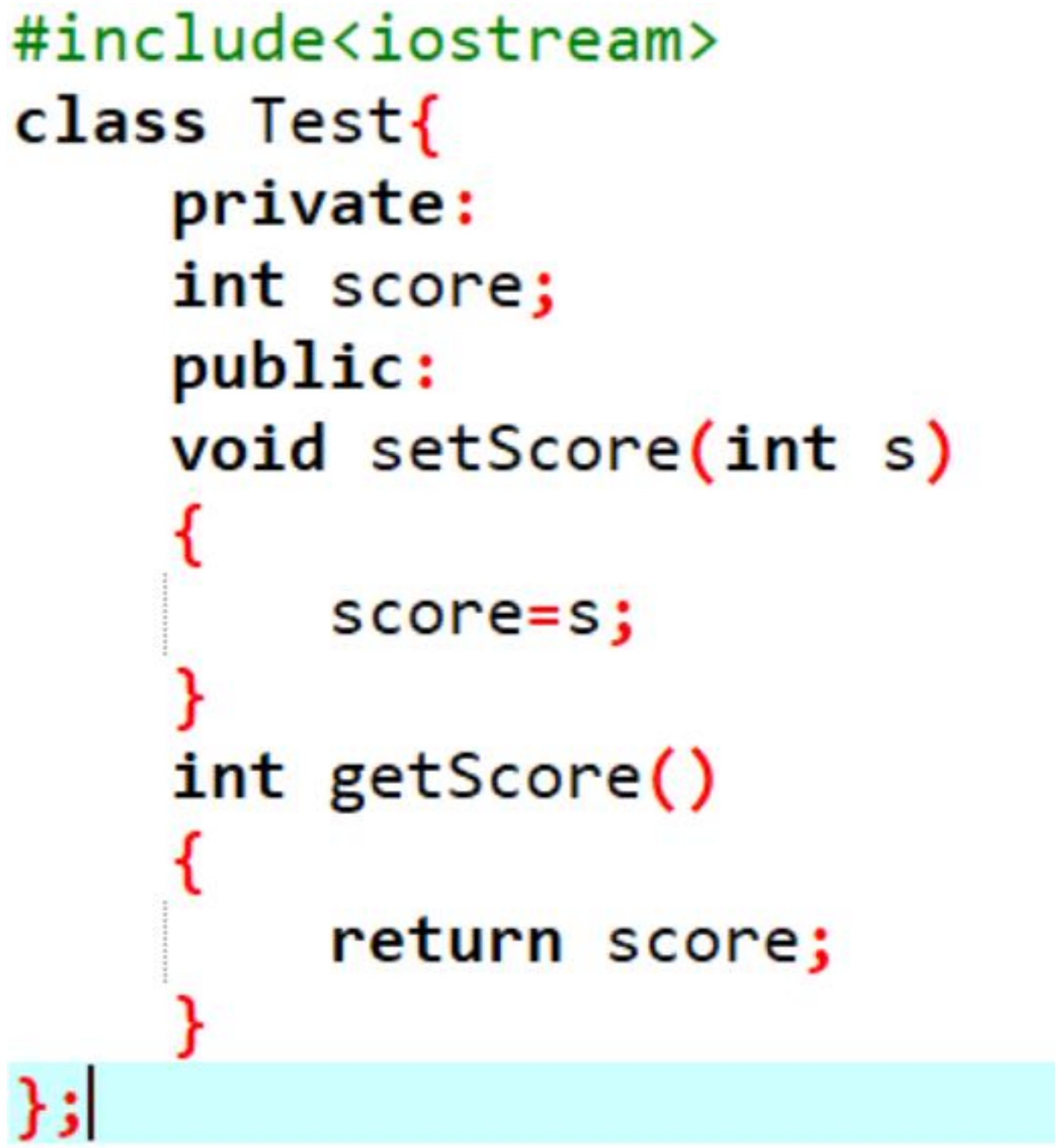
UNIT 2

Shreyasi

CLASSES

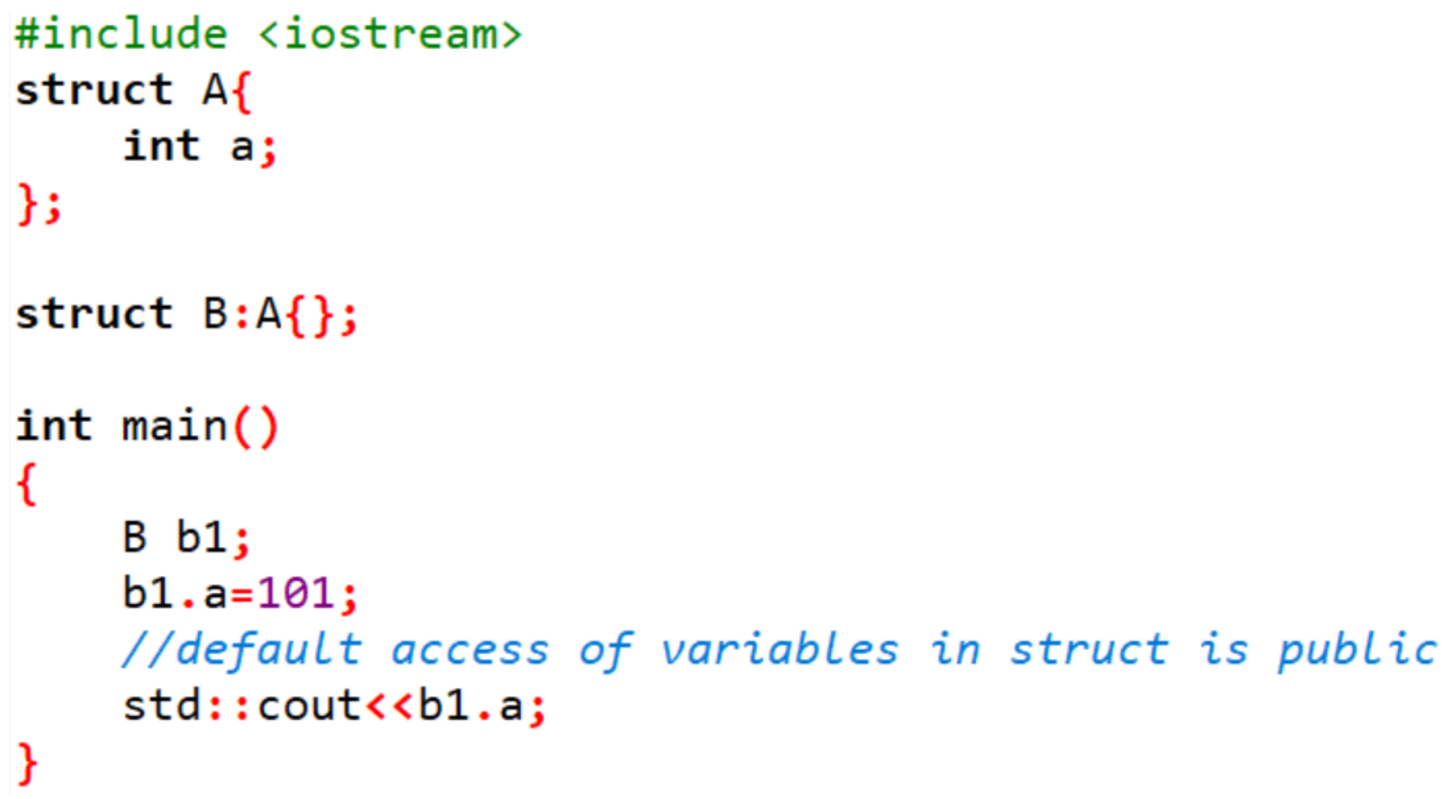


```
#include<iostream>
class Test{
    private:
    int score;
    public:
    void setScore(int s)
    {
        score=s;
    }
    int getScore()
    {
        return score;
    }
};
```

ACCESS SPECIFIERS

- There are three access specifiers in C++:
  - Public – members can be accessed from outside the class.
  - Private – members cannot be accessed from outside the class.
  - Protected – members cannot be accessed from outside the class. They can be accessed in the inherited classes.

CLASS VS STRUCTURE

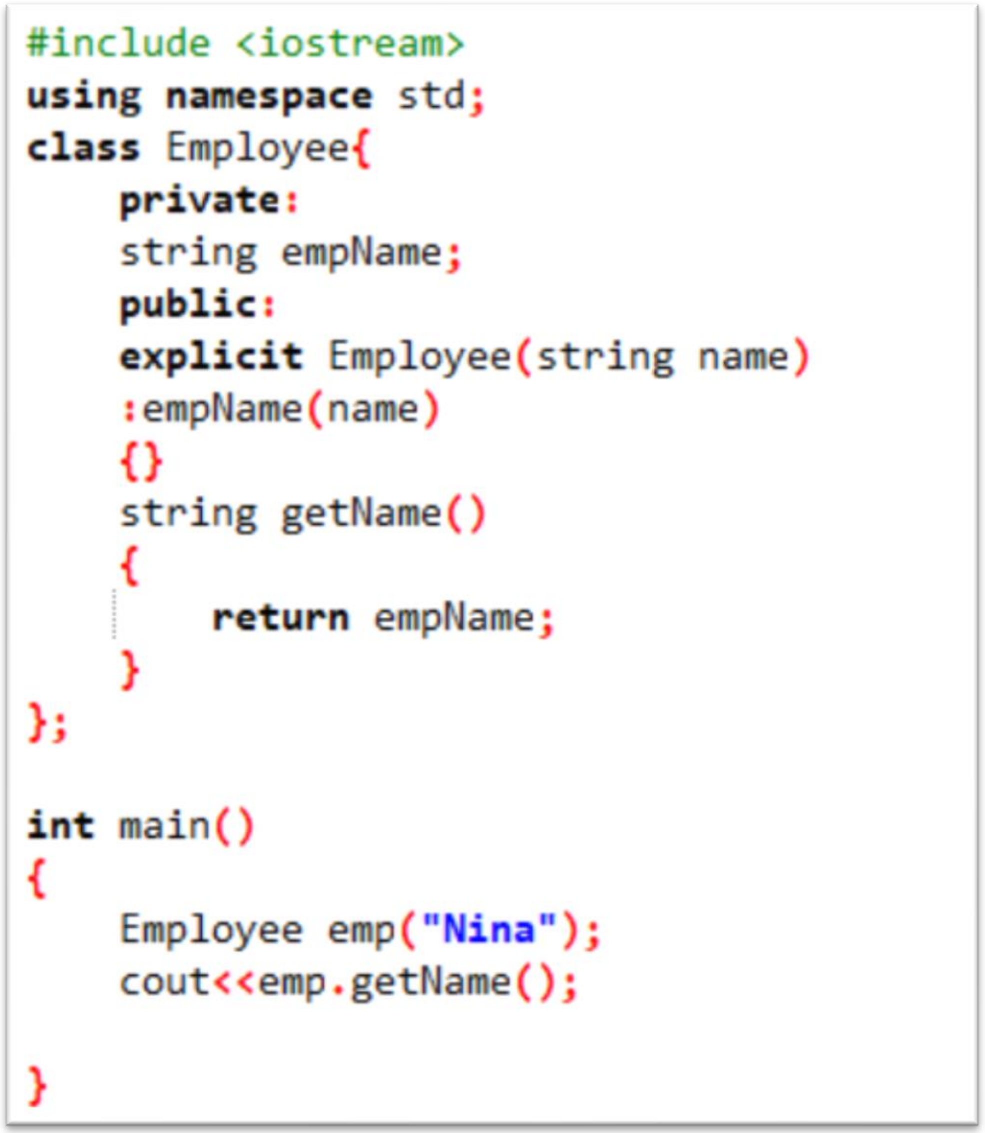


```
#include <iostream>
struct A{
    int a;
};

struct B:A{};

int main()
{
    B b1;
    b1.a=101;
    //default access of variables in struct is public
    std::cout<<b1.a;
}
```

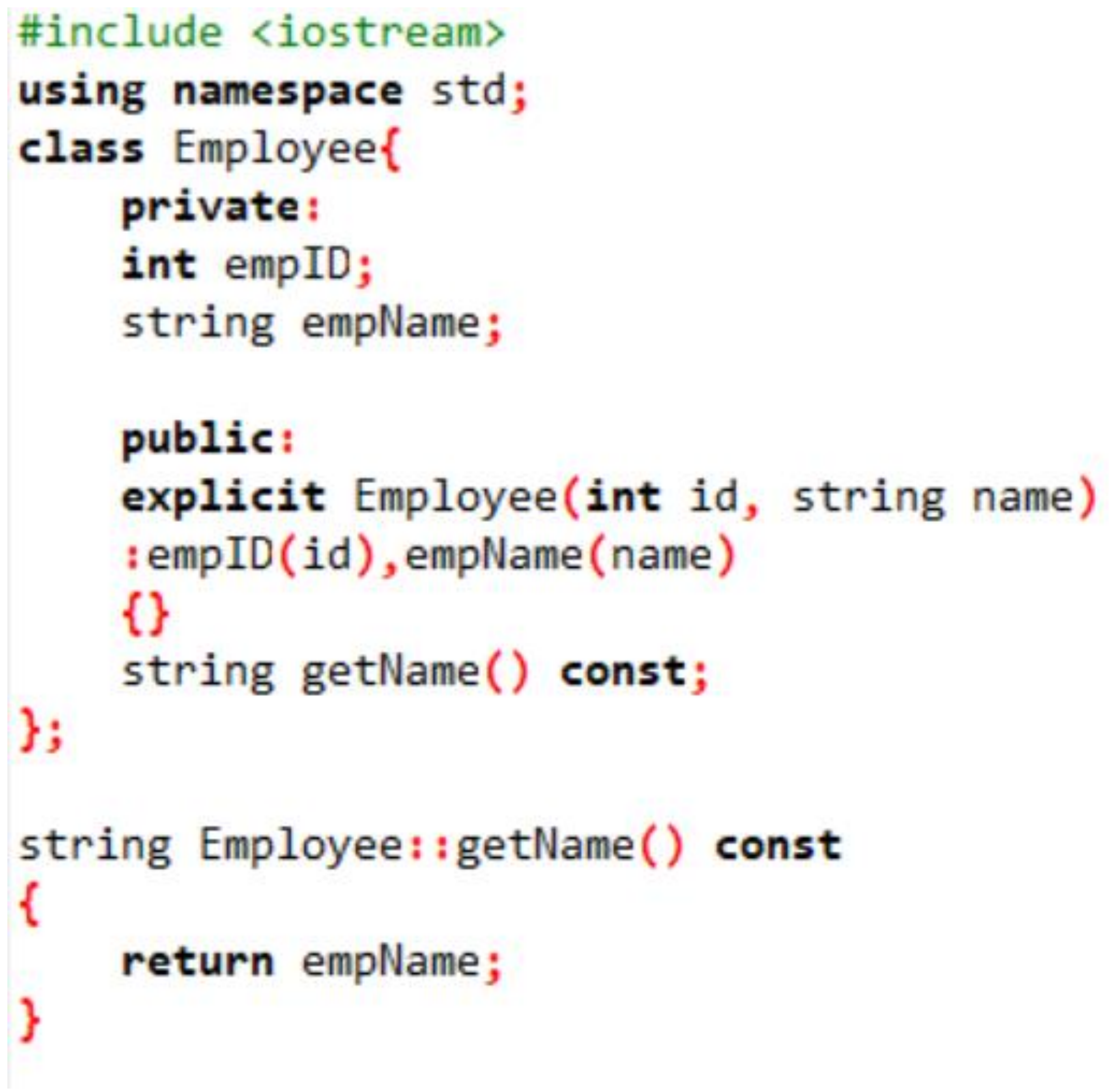
- The default access level of members in a class is private, where as, in a structure it is public.

CONSTRUCTOR

- It is used to initialize an object of the class.
- It is a special member function that should always have the same name as the class.
- It cannot return values, thus it should not have any return data type, including void.
- If a class does not explicitly include a constructor, the compiler provides a default constructor will no parameters.
- If the constructor takes only one argument, it is better to use the explicit keyword to avoid inadvertent type conversions.



```
#include <iostream>
using namespace std;
class Employee{
    private:
    string empName;
    public:
    explicit Employee(string name)
    :empName(name)
    {}
    string getName()
    {
        return empName;
    }
};

int main()
{
    Employee emp("Nina");
    cout<<emp.getName();

}
```



```cpp
#include <iostream>
using namespace std;
class Employee{
    private:
    int empID;
    string empName;

    public:
    explicit Employee(int id, string name)
    :empID(id),empName(name)
    {}
    string getName() const;
};

string Employee::getName() const
{
    return empName;
}
```