Features of OOP(Classes and Objects)

Md. Alamgir Hossain

Senior Lecturer,

Prime University

Mail: *alamgir.cse14.just@gmail.com*

Classes and Objects

- ➢**Class:** A class in C++ is the building block, that leads to Object-Oriented programming.
- ✓It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.
- ✓A C++ class is like a blueprint for an object.
- ✓**Example:** Consider the Class of **Cars**. There may be many cars with different names and brand but all of them will share some common properties like all of them will have *4 wheels*, *Speed Limit*, *Mileage range* etc. So here, Car is the class and wheels, speed limits, mileage are their properties.

Classes and Objects

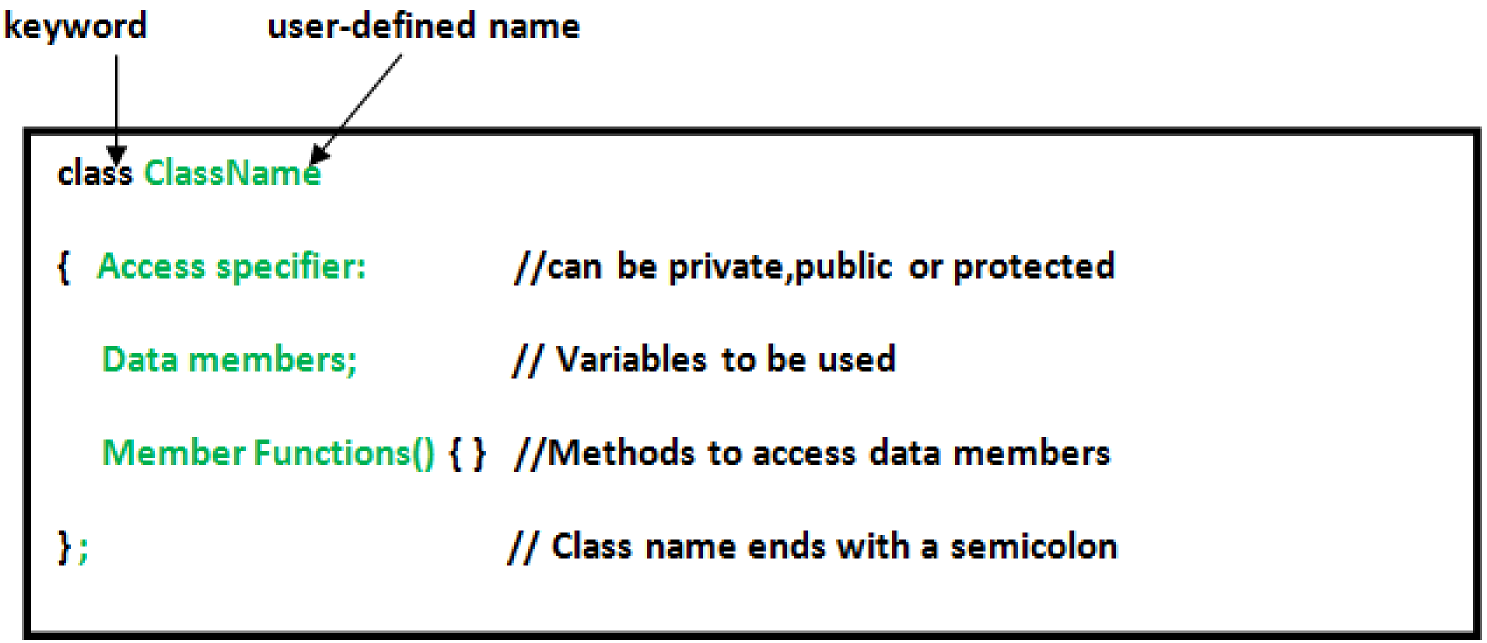
➢ **Object** is an instance of a Class.

✓ When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

✓ **Example:** Honda, Toyota, BMW(Bayerische Motoren Werke) etc.

Defining Class and Declaring Objects

- A class is defined in C++ using keyword class followed by the name of class. The body of class is defined inside the curly brackets and terminated bv a semicolon at the end.



```
class ClassName

{  Access specifier:          //can be private,public or protected

   Data members;              // Variables to be used

   Member Functions() { }     //Methods to access data members

};                            // Class name ends with a semicolon
```

Defining Class and Declaring Objects

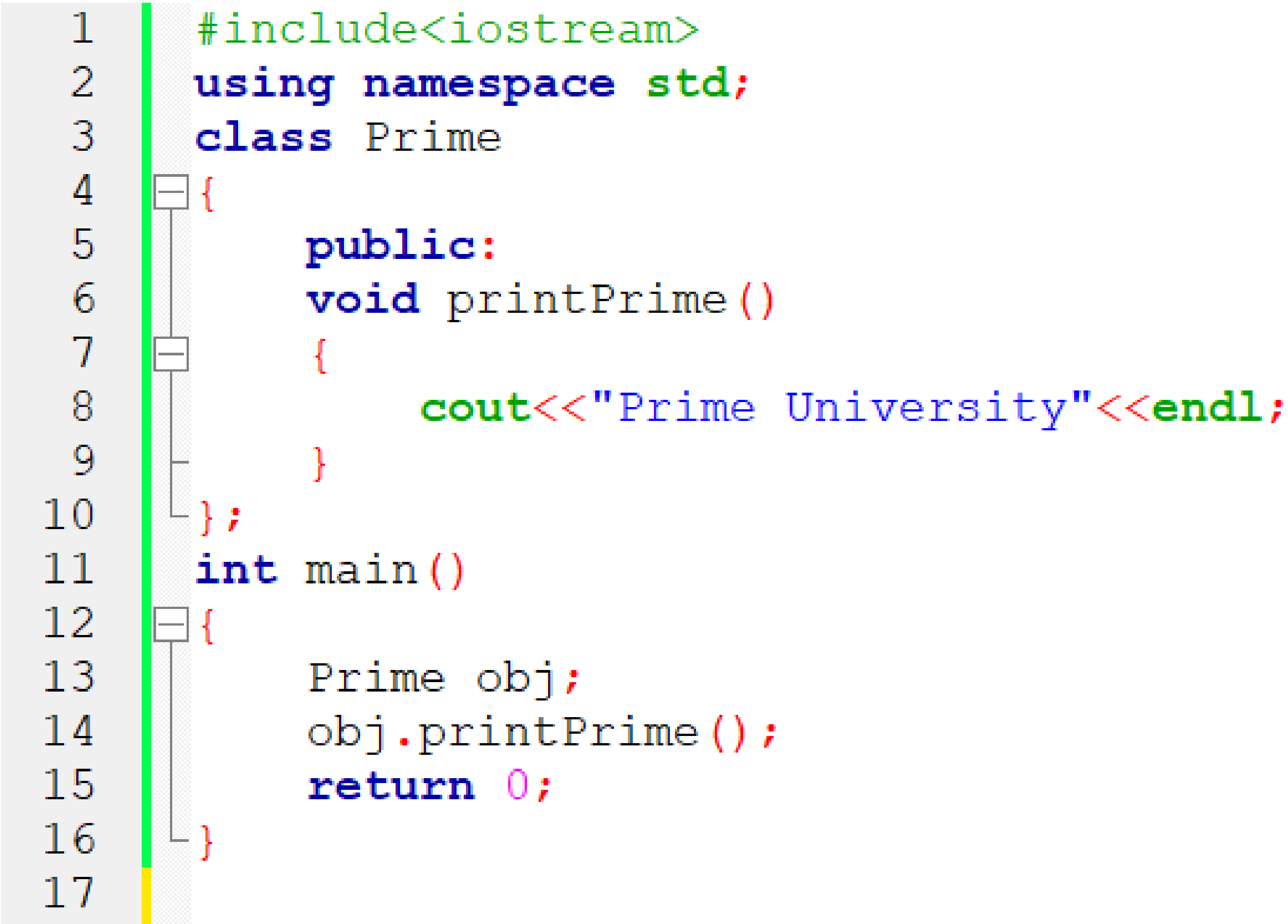
- **Declaring Objects:** When a class is defined, only the specification for the object is defined; no memory or storage is allocated.
- To use the data and access functions defined in the class, you need to create objects.

✓ **Syntax: ClassName ObjectName**

Accessing Data Members and Member Functions

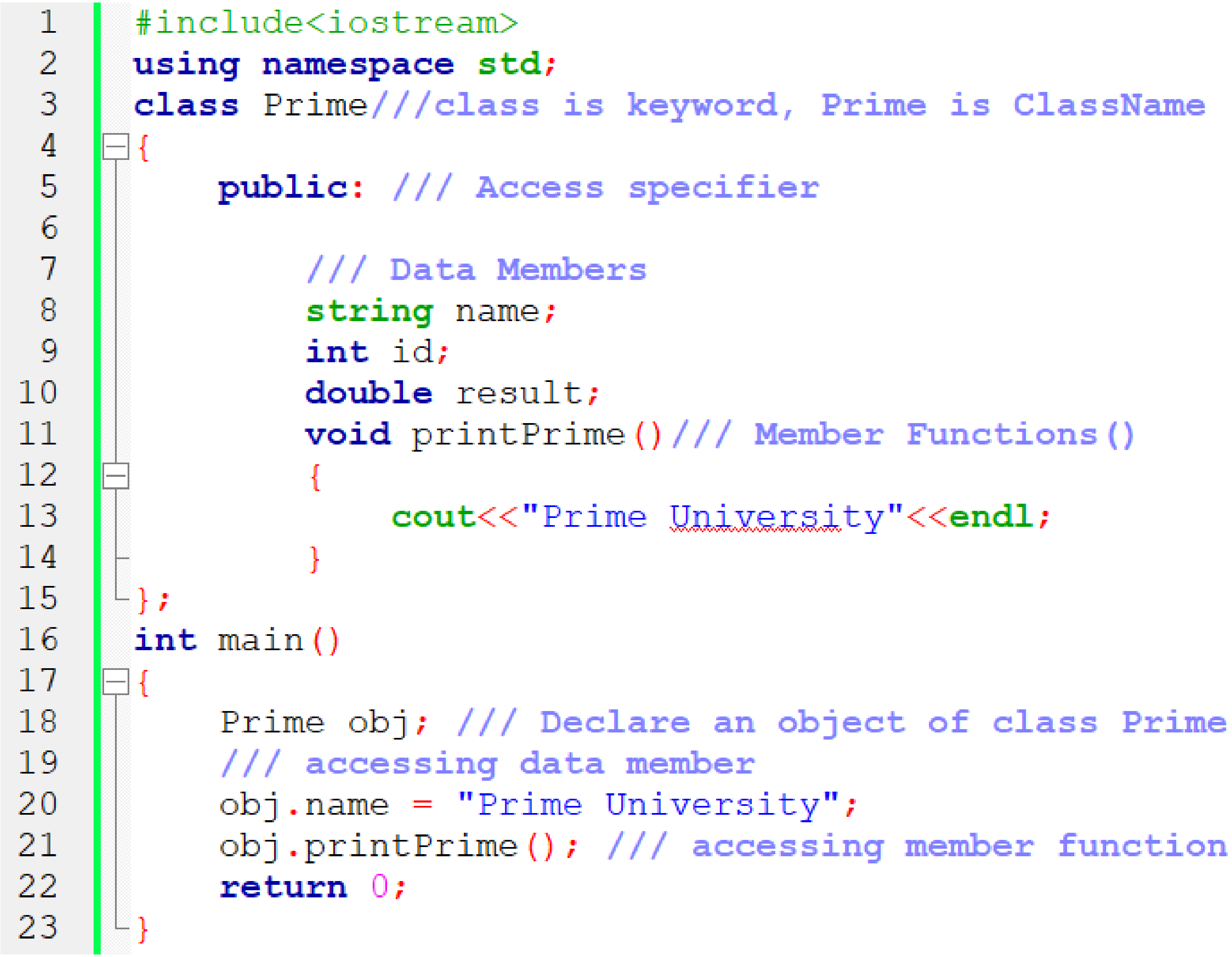
- The data members and member functions of class can be accessed using the dot(‘.’) operator with the object.
- For example if the name of object is *obj* and you want to access the member function with the name *printName()* then you will have to write *obj.printName()* .
- Accessing a data member depends solely on the access control of that data member.
- There are 3 types of access modifiers available in C++: **Public**, **Private**, **Protected**

Defining Class and Declaring Objects



```cpp
#include<iostream>
using namespace std;
class Prime
{
    public:
    void printPrime()
    {
        cout<<"Prime University"<<endl;
    }
};
int main()
{
    Prime obj;
    obj.printPrime();
    return 0;
}
```

Defining Class and Declaring Objects



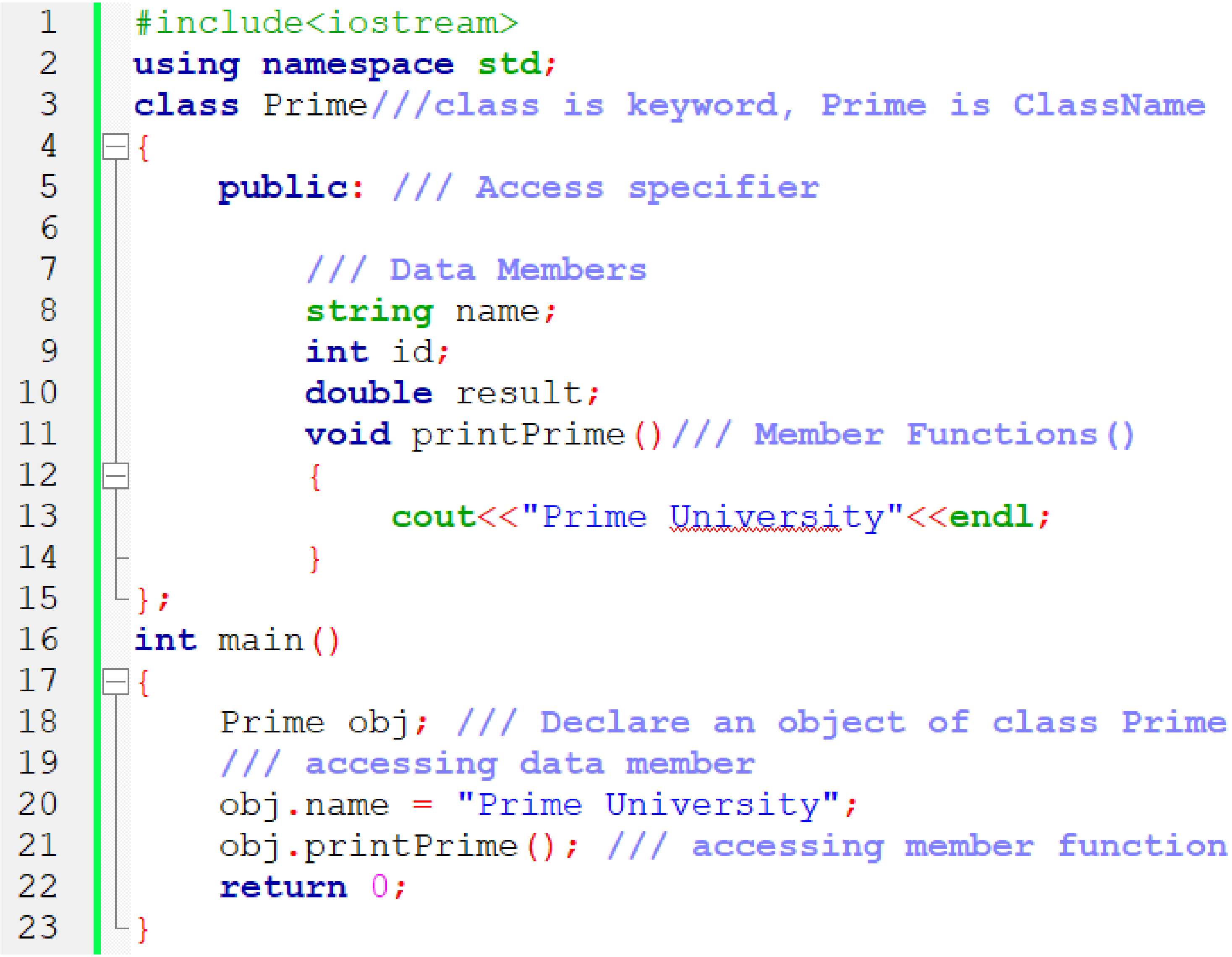
```
#include<iostream>
using namespace std;
class Prime///class is keyword, Prime is ClassName
{
    public: /// Access specifier

        /// Data Members
        string name;
        int id;
        double result;
        void printPrime()/// Member Functions()
        {
            cout<<"Prime University"<<endl;
        }
};
int main()
{
    Prime obj; /// Declare an object of class Prime
    /// accessing data member
    obj.name = "Prime University";
    obj.printPrime(); /// accessing member function
    return 0;
}
```

Member Functions in Classes

- There **are 2 ways** to define a member function:
- Inside class definition, Outside class definition
- To define a member function outside the class definition we have to use the scope resolution :: operator along with class name and function name.

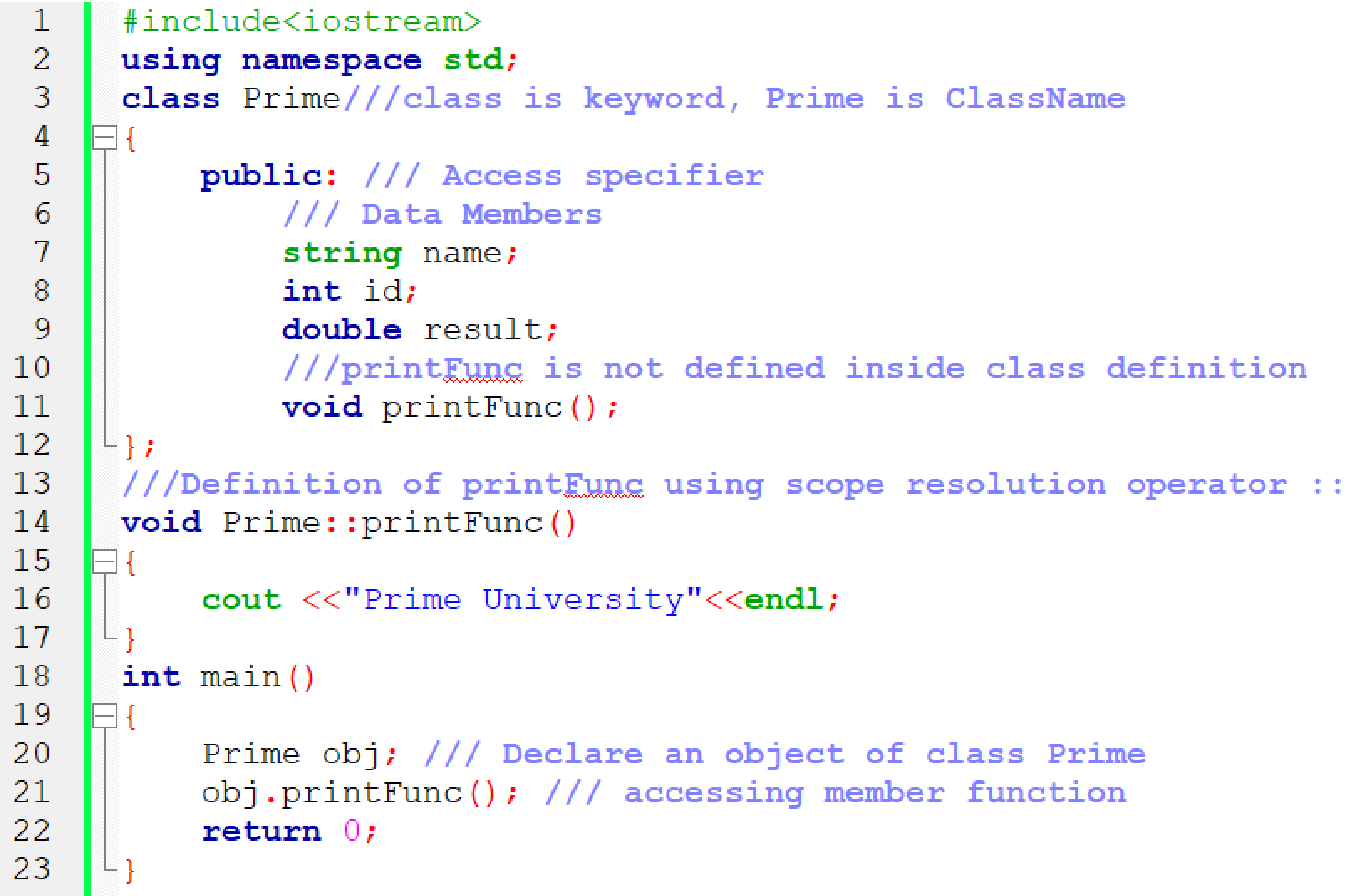
Member Functions in inside Class



```cpp
#include<iostream>
using namespace std;
class Prime///class is keyword, Prime is ClassName
{
    public: /// Access specifier

        /// Data Members
        string name;
        int id;
        double result;
        void printPrime()/// Member Functions()
        {
            cout<<"Prime University"<<endl;
        }
};
int main()
{
    Prime obj; /// Declare an object of class Prime
    /// accessing data member
    obj.name = "Prime University";
    obj.printPrime(); /// accessing member function
    return 0;
}
```

Member Functions in outside Class



```cpp
#include<iostream>
using namespace std;
class Prime///class is keyword, Prime is ClassName
{
    public: /// Access specifier
        /// Data Members
        string name;
        int id;
        double result;
        ///printFunc is not defined inside class definition
        void printFunc();
};
///Definition of printFunc using scope resolution operator ::
void Prime::printFunc()
{
    cout <<"Prime University"<<endl;
}
int main()
{
    Prime obj; /// Declare an object of class Prime
    obj.printFunc(); /// accessing member function
    return 0;
}
```

Thank You