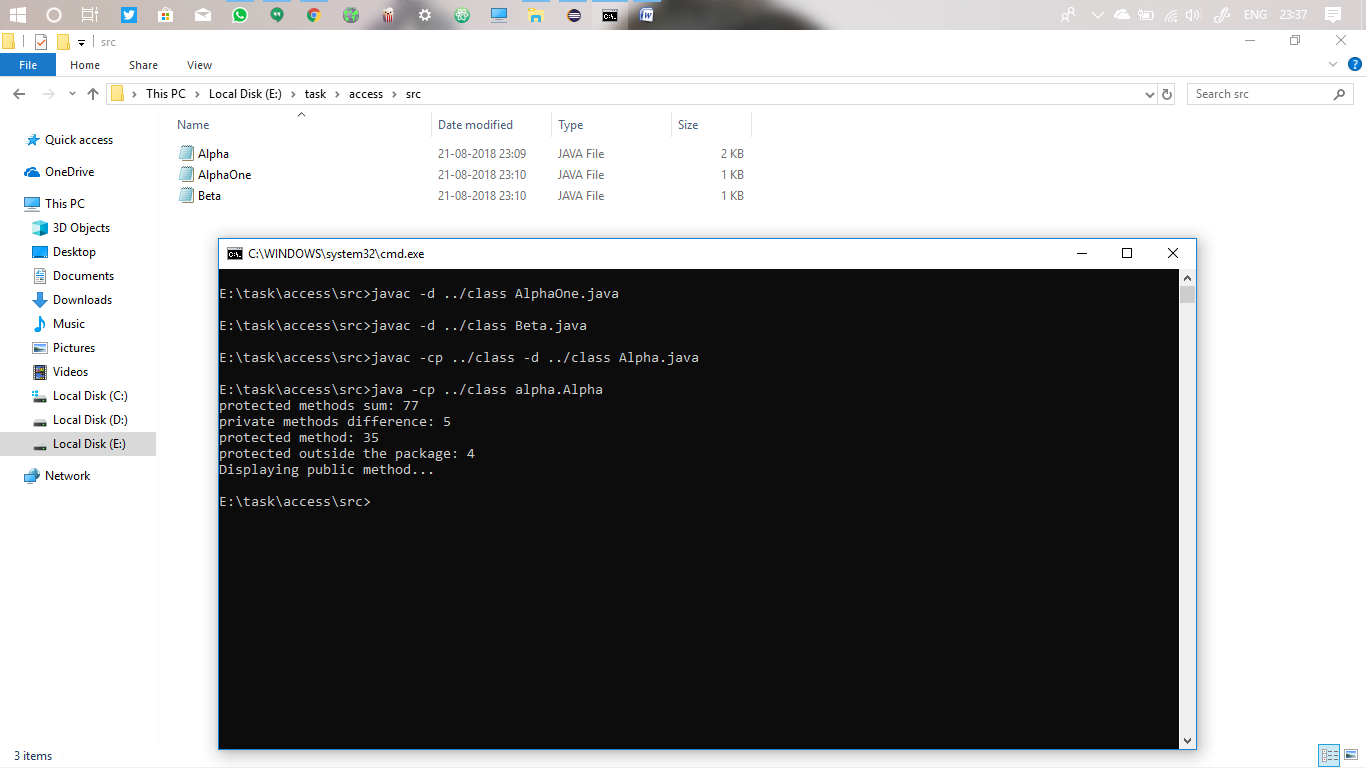
**Access Specifiers:**

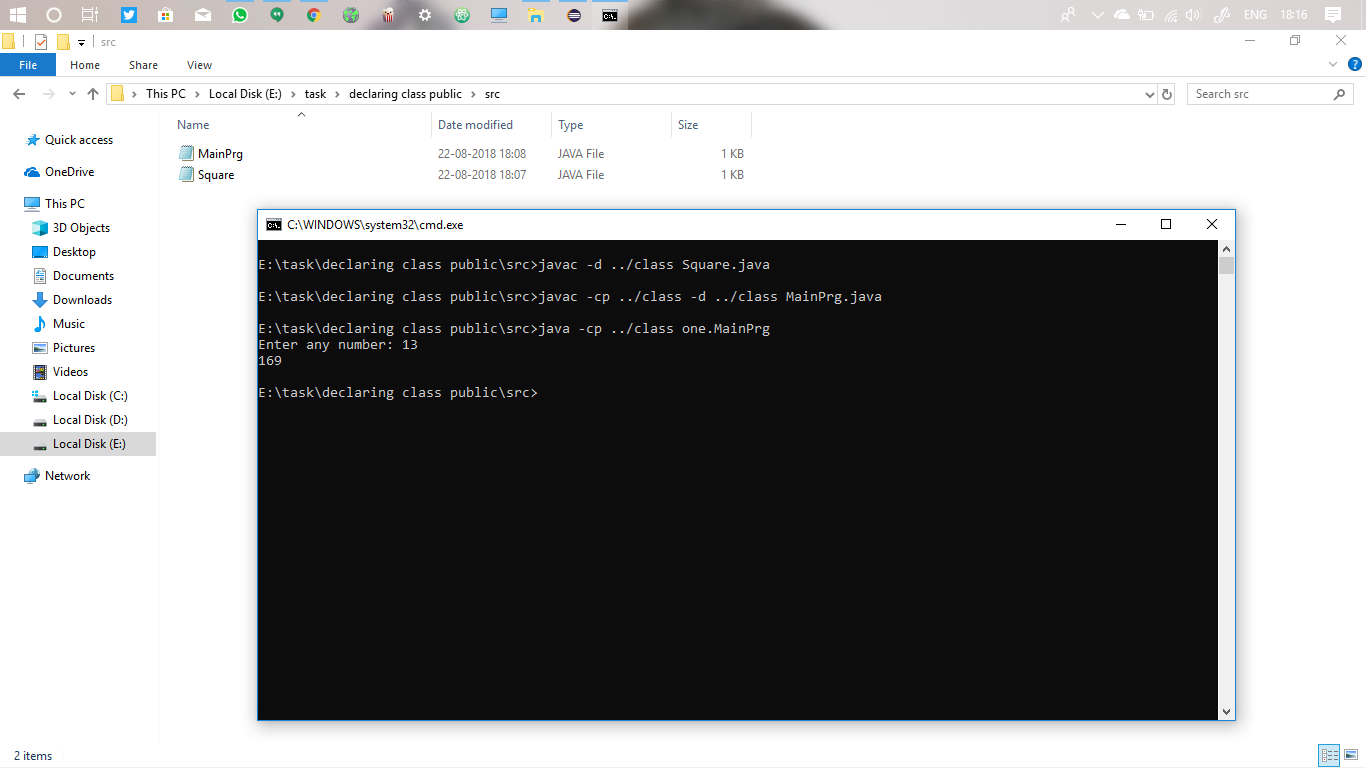
Access specifiers are keywords that sets the visibility of variables, methods, and class from the main function. There are access specifiers like public, private and protected also default sometimes. The public modifier can be accessed anywhere within or outside the class or even the package, the protected can be accessed within the package with help of inheritance, the private modifier can be accessed only within the same class.



Here I wrote all private, protected and public methods and accessed them.

**Declaring class as public**

By making the class as public, it can be accessed anywhere even outside the package. The public specifier makes the class visible everywhere.



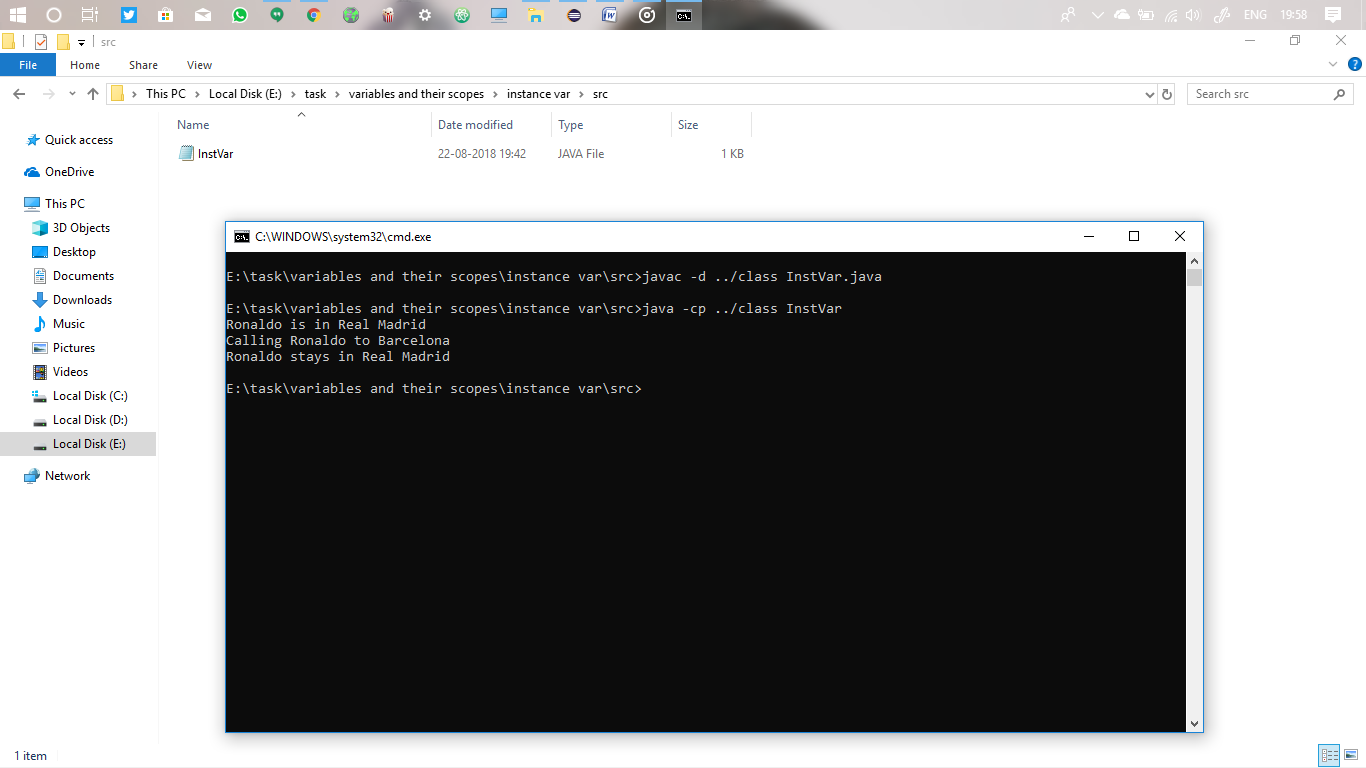
First a sub class is made public and stored in a package. Next the main class is written in some other package. As the sub class is made public, by importing the package they can be accessed even outside its package.

**Variables and their scopes:**

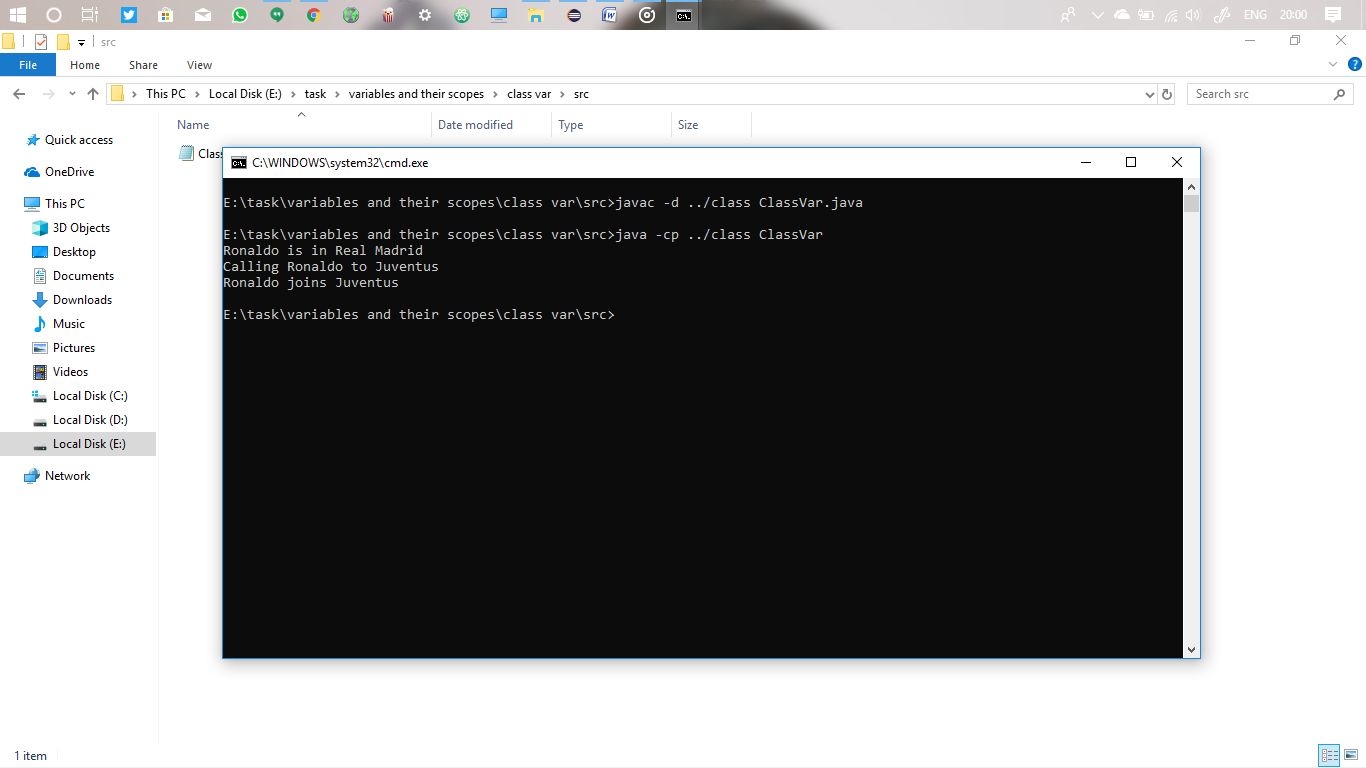
There are instance variables and class variables.

The class variables are also called static variables. Once it has been created any changes made in any object, reflects to every other place just like call by reference, that’s how I remember this.

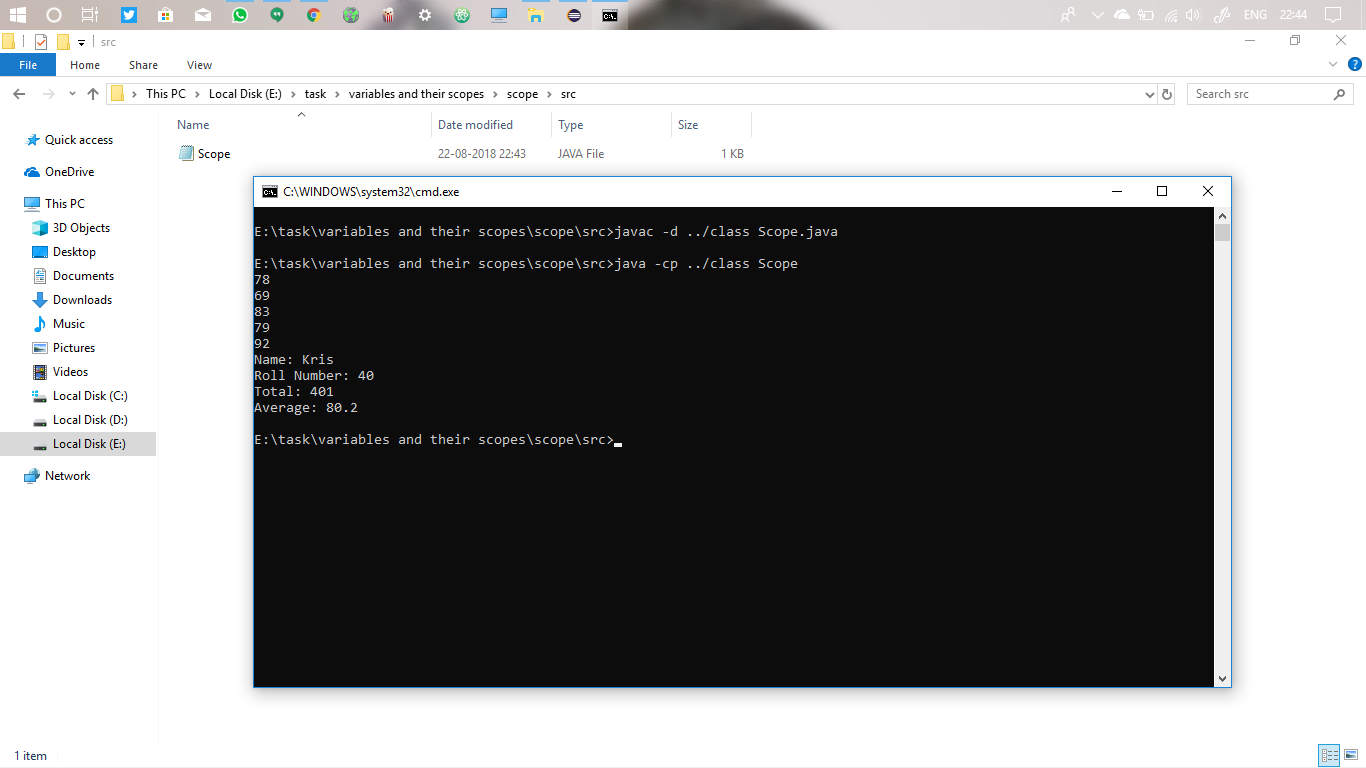
But the instance variables doesn’t change for every objects instead they remains the same for other objects. Not only the objects but also for the methods.

There are different scopes based on how we use them. Local scopes are one that can be used only within the block. It can’t be accessed outside. Loop scopes, method scopes are some of them. And the global scope variable can be accessed anywhere within the class called class level scope. 

The above print is instance variable which doesn’t changes for every other call.



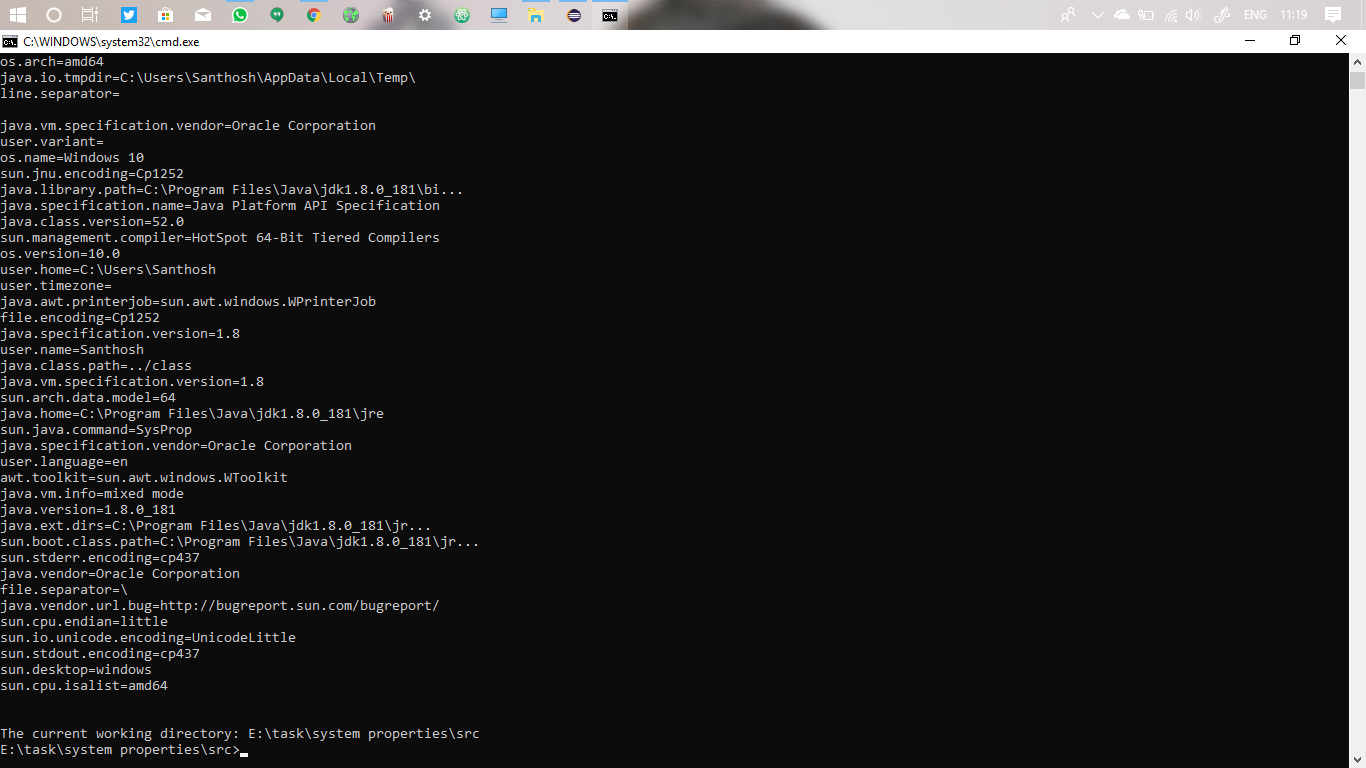
Now the above given is class variable which I declared in static



Here I declare some of the variables within class scope, method scope and main.

**System Properties**

The Properties of the System can be accessed from the java Properties class.

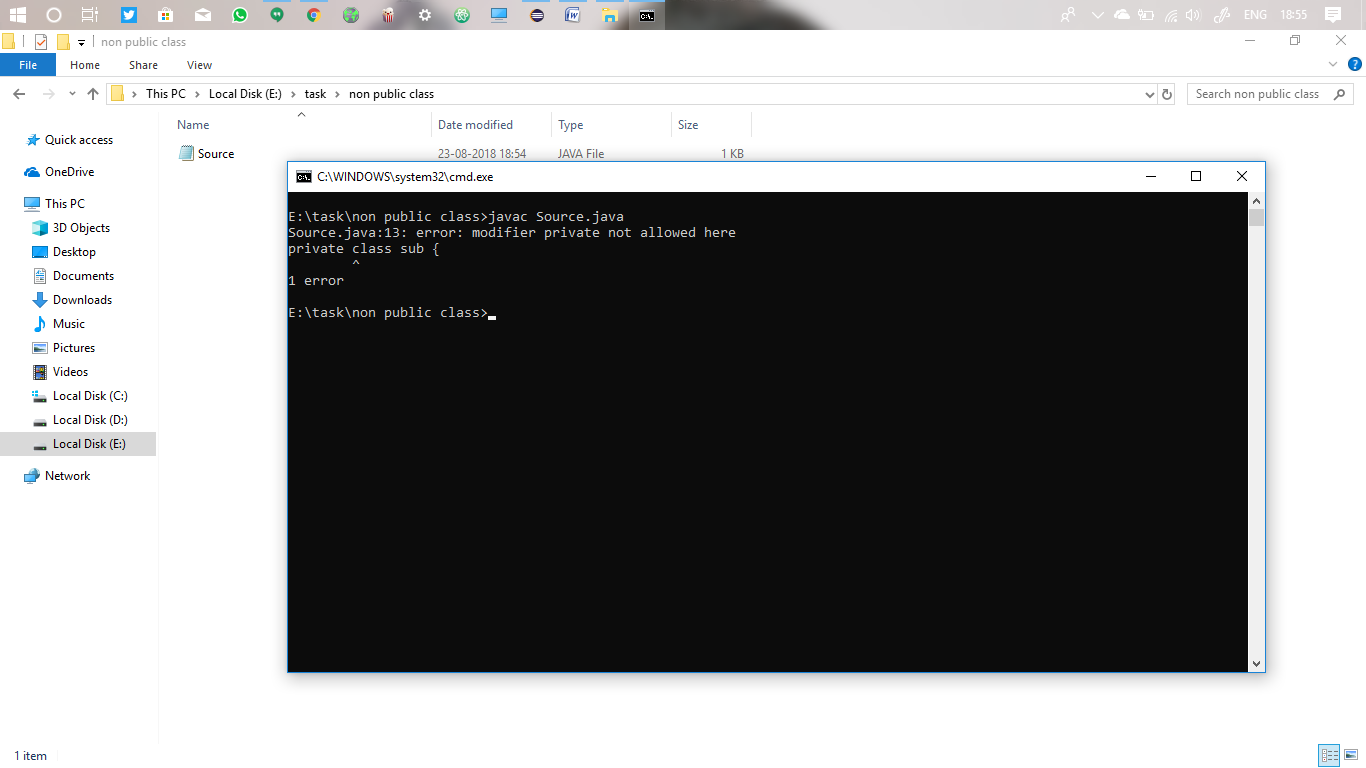


**Main method**

The main method belongs to the class in which you have defined the method main. Just for the above program system properties, SysProp is the class to which the main() method is belonged, and the way to access this main() method is public. This method can be accessed from anywhere outside the package, that’s how JVM access the body. If it is not a public class the java virtual machine can’t access the main and can’t enter into the program. So it says the main() is not visible for JVM.

So for each program the main() method belongs to the class itself and the way to access them is public.

**Non public classes**

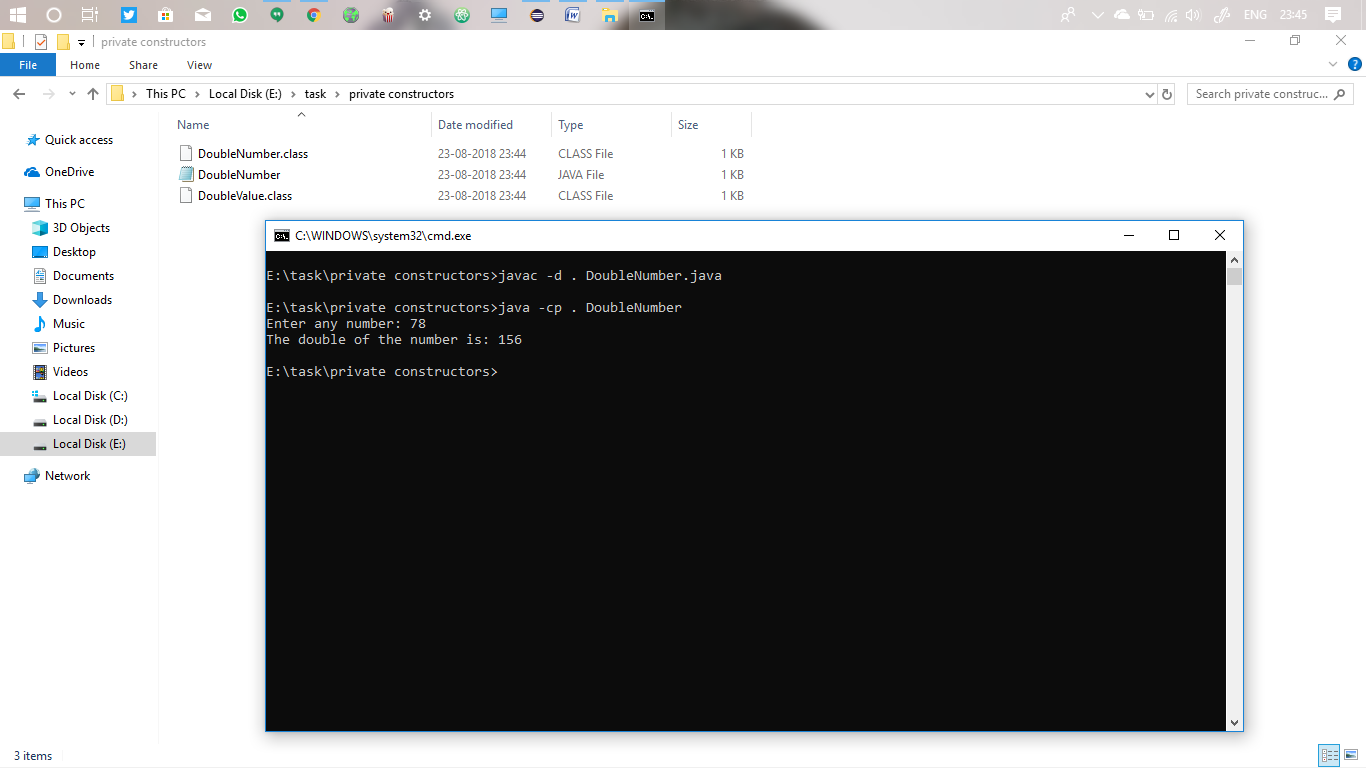
Top level classes must always be declared as public. Its waste to declare the class as private, as no one can access them. 

But the nested class such as the class inside the class can be declared as private.

Also the top level can’t be protected, only the inner class can be protected, so that we can access the inner class from the outer class and the outer class from the main.

**Private constructors**

Anything that is declared as private cannot be accessed outside the class. When the constructor is declared as private the whole object creation of that class becomes invisible. But we can access them by creating a static method and object inside that class, as for the static method we don’t need object to call that method.

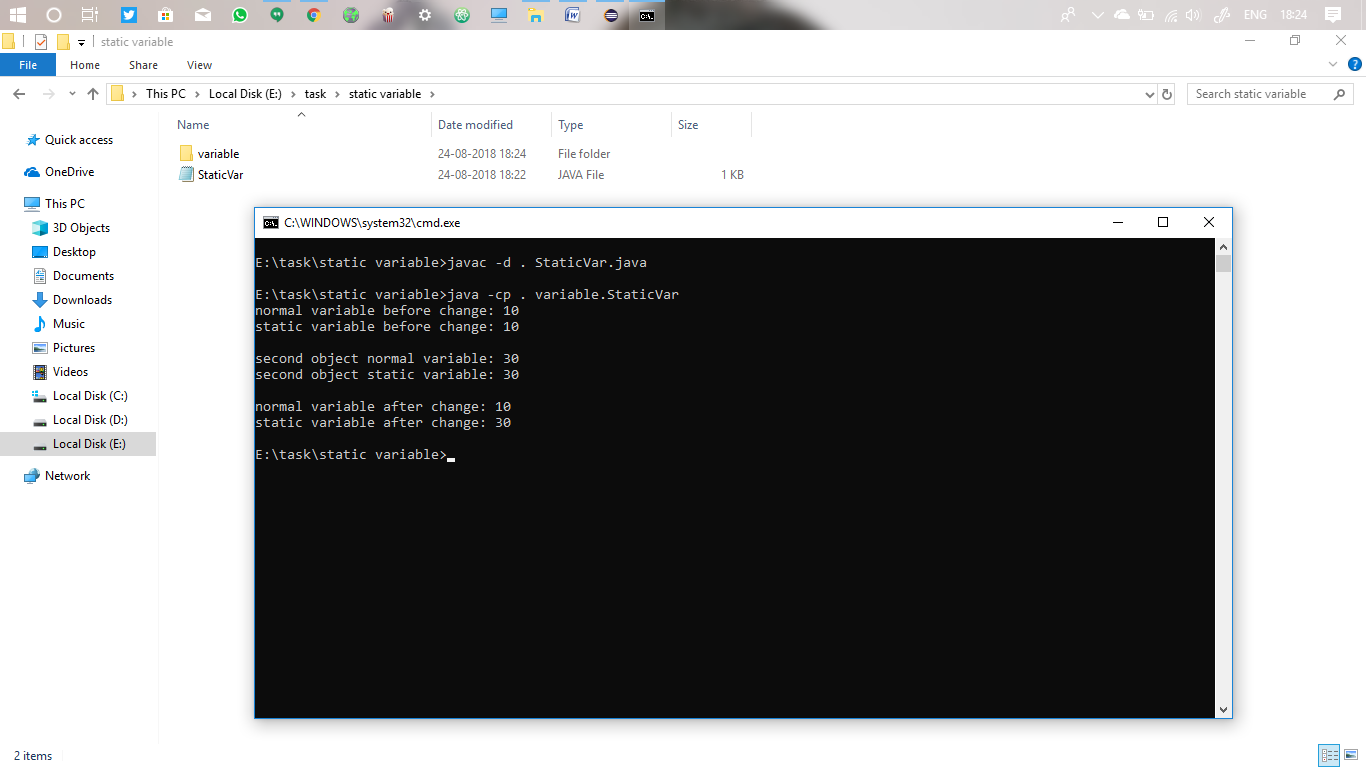
****

The main advantage of this private constructor is Singleton class. We can restrict the object creation.

By declaring the object **null** inside the singleton class and in the static method we can assign the object creation only if that previous is **null.**

**Static variable**

Static variable is a variable that is common to all instance of a class. It is used when only one copy of the variable is needed to use. It is said as a class variable. When one copy of that variable is changed, all the other copies changes.

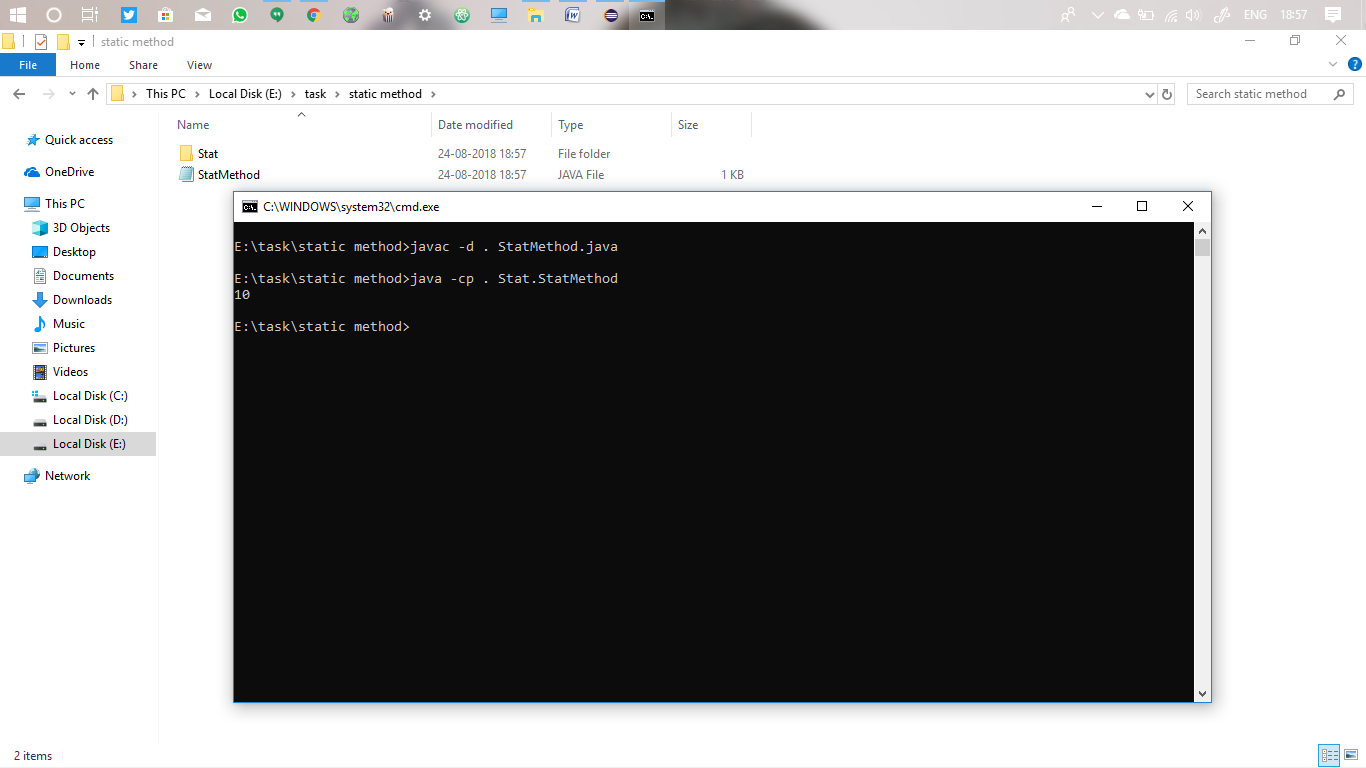
****

**Static method**

The static methods belongs to the class itself rather than its object.

They can only access the static variable, and accessing these methods doesn’t need any object to be created.

With just the class name we can access the static method.

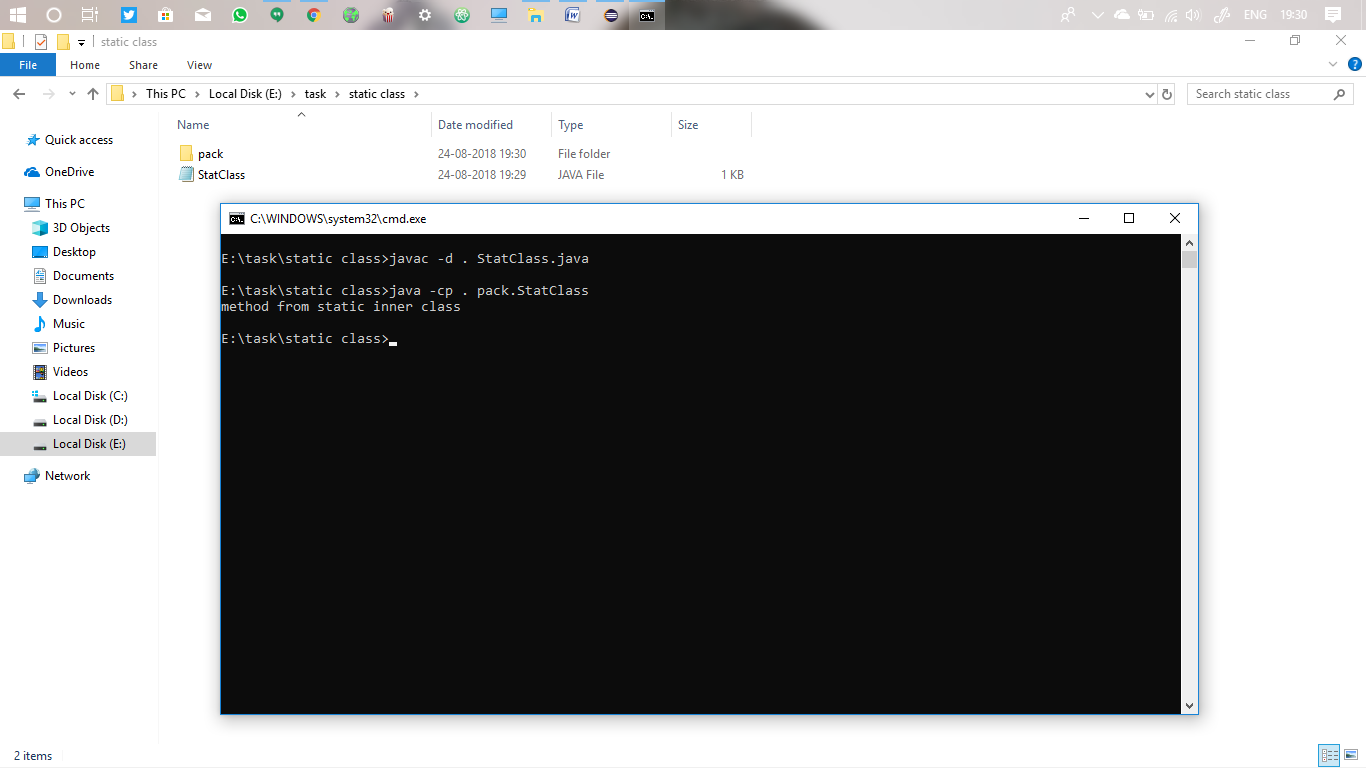


**Static class**

Just like variables and methods, classes can also be declared as static. But the outer or main class can’t be declared as static, only the inner nested class we can.

And with the outer class object we can access the methods from inner static class.

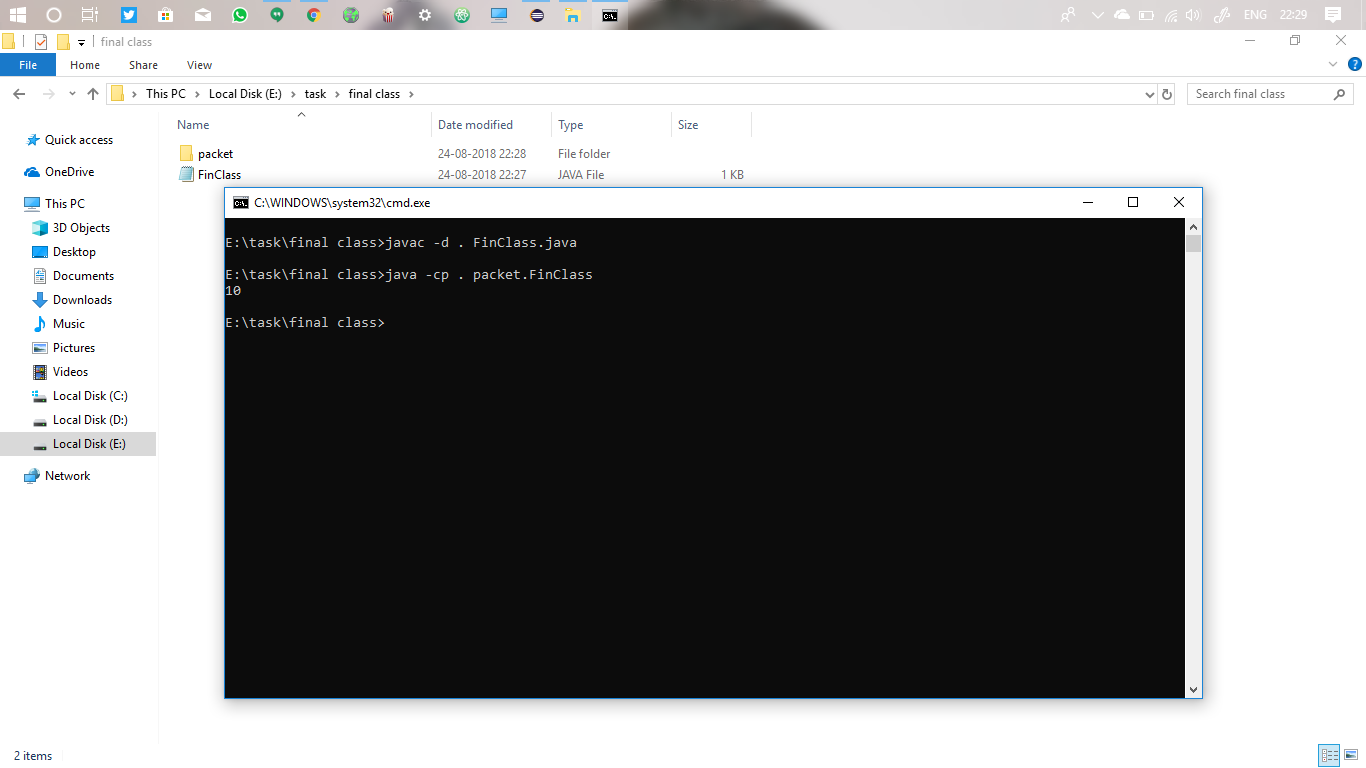
Also these static classes can only access the variables and methods outside which are only static.

****

**Final Class**

Final is a keyword used to make the variable assign the value only once. It can be assigned to a variable or method or even class.

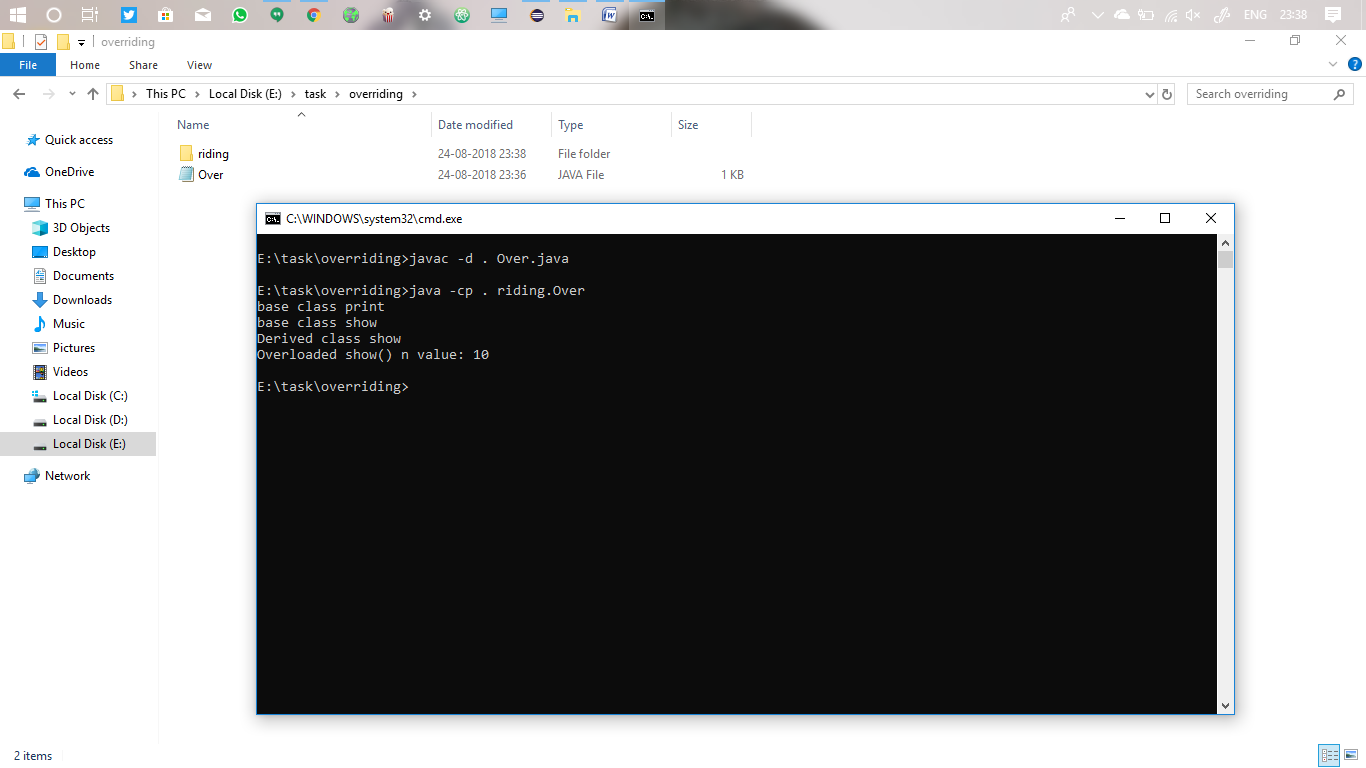
When a final keyword is assigned to a class, their values and methods can’t be assigned to another class. That is this class is final and we can’t inherit their values outside. Thus we can restrict the inheritance of methods and variables by this method.

****

**Overriding and overloading**

Overriding can be prevented by using final keyword to the methods. As final is given to a method we can inherit the method from the base class to the child thereby overriding is prevented. We can also use static and private too. These too prevent the accessibility of the method by the derived classes.

But overloading, it is unusual to prevent the occurrence. It’s just keeps the same name of the method but with different number of arguments or argument types. We can’t prevent the overriding of methods.



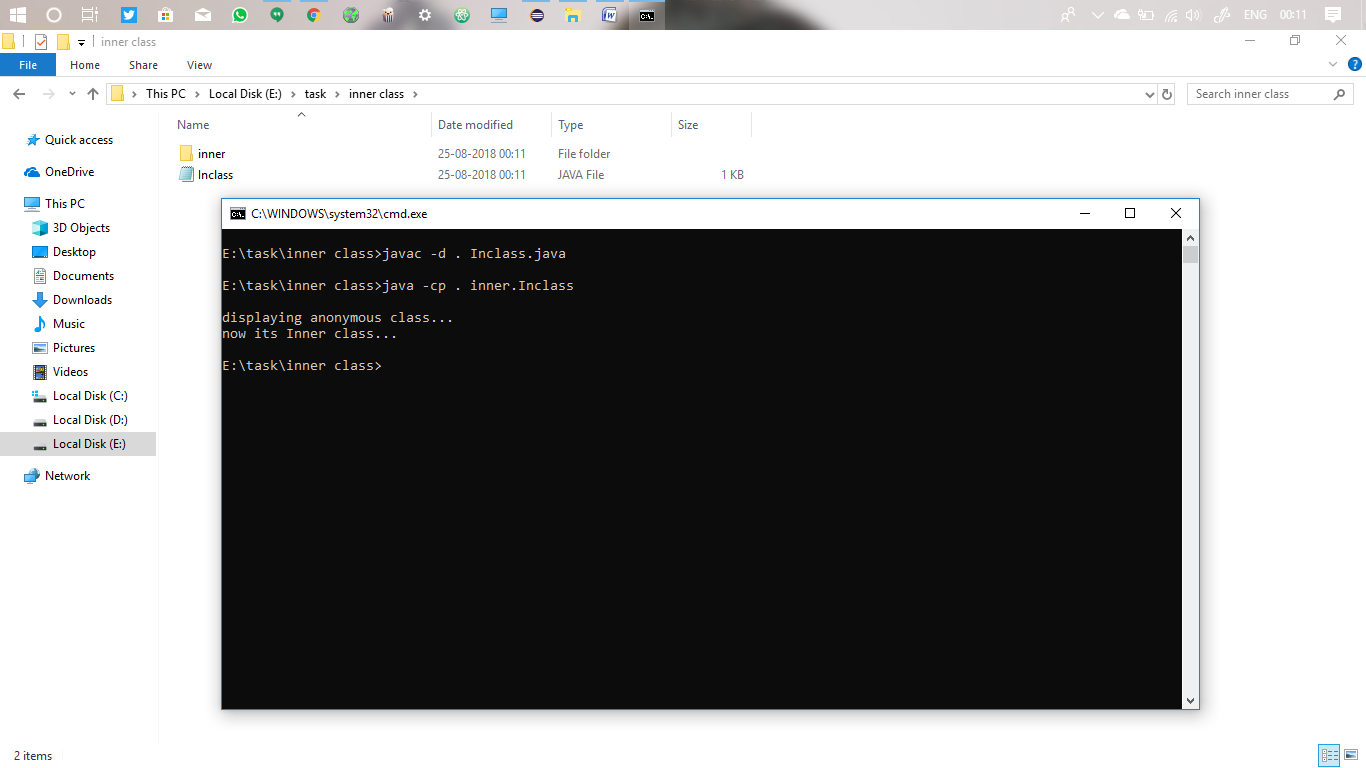
Here we can’t inherit the final print() method. Also we can see here I used the show() function twice, one without argument and one with an argument.

**Inner class and anonymous class**

When a class is nested inside another class then it is called inner class.

Anonymous class will not have any name, and only one object can be created. Such as to access an abstract class we can use anonymous inner class. When we create an instance of a sub class or abstract we can override its method within the object. Thus formed class is called anonymous inner class.

It is useful in grouping a type of class or when private methods of a class are needed to be accessed we can create a class within the class, and now this inner class can access these private methods saying that class is private.



The difference between the inner and nested class is that it is almost same but that nested classes are static and the inner classes are non static. Thus the static class can access only the static members of the outer class thus without creating any instance of that class we can access those methods.