Week-7

Name – Randrita Sarkar [11500219058]

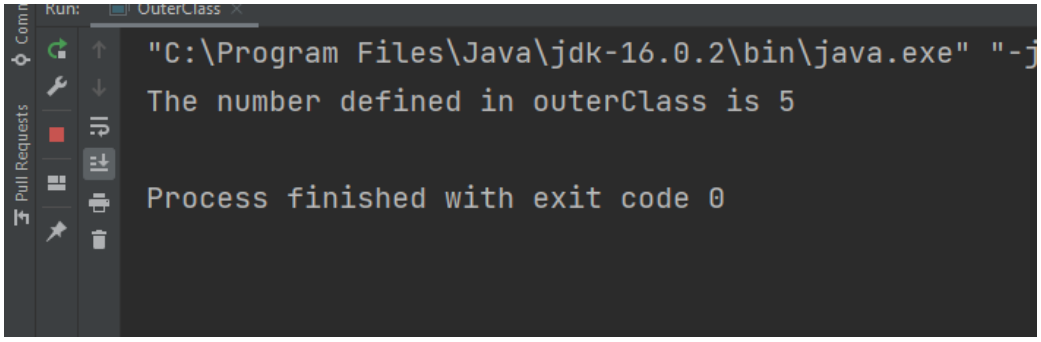
IT PCC-CS593 L - OBJECT ORIENTED PROGRAMMING LAB

Inner Class

Show that an inner class has access to the **private** elements of its outer class. Determine whether the reverse is true

```
package com.randrita.week7;

/*Show that an inner class has access to the
private elements of
its outer class. Determine whether the reverse is
true*/

public class OuterClass {

    private int k=5;
    class InnerClass{
        void display(){
            System.out.println("The number defined
in outerClass is "+ k);
        }
    }
    void show(){
        InnerClass inner = new InnerClass();
        inner.display();
    }

    public static void main(String[] args) {
       OuterClass number = new OuterClass();
       number.show();

    }
}
```

Output:

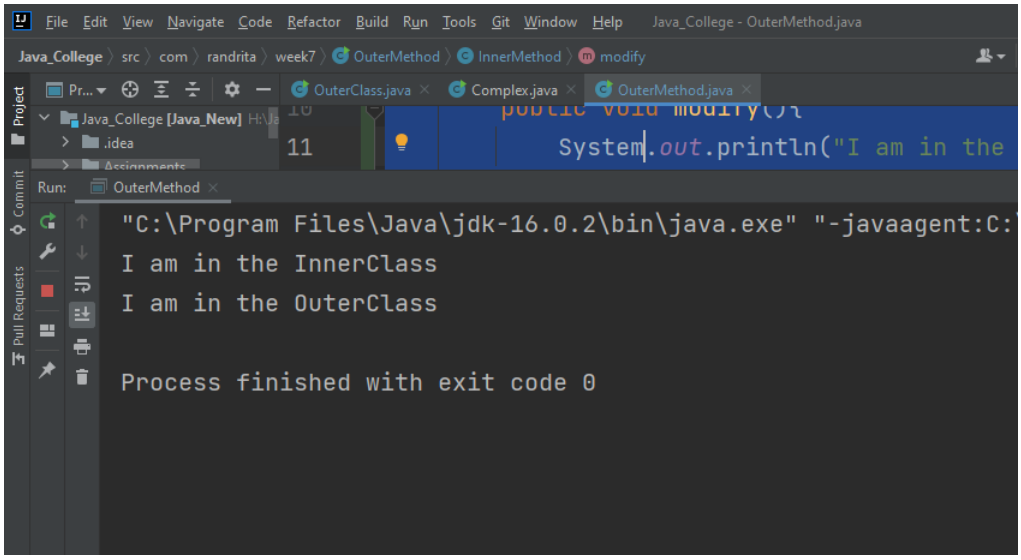


2. Create an inner class with a method that modifies the outer class field and calls the outer class method. In a second outer class method, create an object of the inner class and call its method, then show the effect on the outer class object.

```
package com.randrita.week7;

public class OuterMethod {

    public void modify(){
        System.out.println("I am in the
OuterClass");
    }

    class InnerMethod{
        public void modify(){
            System.out.println("I am in the
InnerClass");
        }
    }

    public void display(){
        InnerMethod inner = new InnerMethod();
        inner.modify();
    }

    public void outerDisplay(){
        modify();
    }

    public static void main(String[] args) {
         OuterMethod outer = new OuterMethod();
```

```
            outer.display();
            outer.outerDisplay();
        }
}
```

Output:

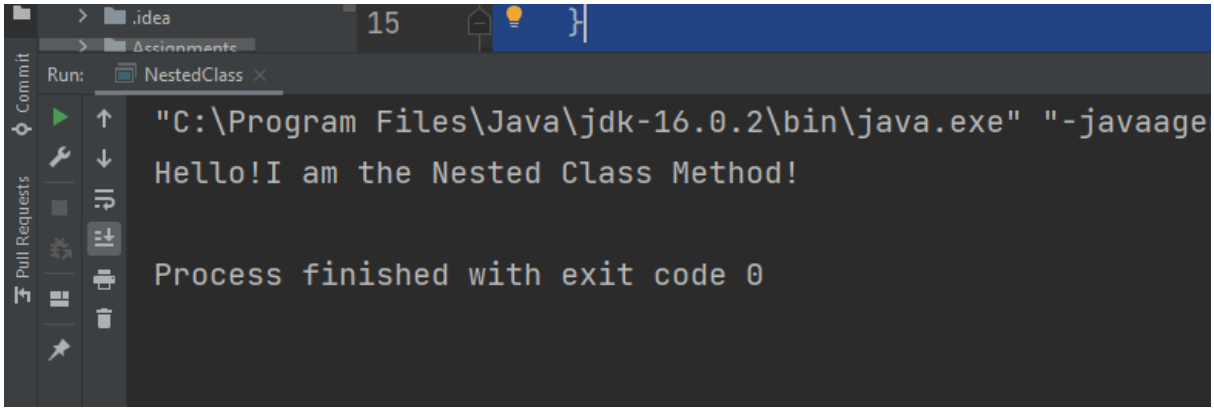


3. Create a class containing an inner class that itself contains an inner class. Repeat this using nested classes. Note the names of the .class files produced by the compiler.

```
package com.randrita.week7;

class Outer {
    class Inner{
        class NestedInner{
            void display(){
                System.out.println("Hello!I am the
Nested Class Method!");
            }
        }

    }
    void display(){
        Inner.NestedInner variable = new
Inner().new NestedInner();
        variable.display();
```

```
    }
}

public class NestedClass {
    public static void main(String[] args) {
        Outer variableOuter = new Outer();
        variableOuter.display();

    }
}
```

Output:



4. Create a class with an inner class that has a nondefault constructor (one that takes arguments). Create a second class with an inner class that inherits from the first inner class.

```
package com.randrita.week7;

/*Create a class with an inner class that has a
non default constructor (one that takes
arguments). Create a
second class with an inner class that inherits
from the first inner class.*/

public class InnerInherit extends
OuterInheritedSecondClass {
    public static void main(String[] args) {
        OuterInheritedSecondClass secondInner =
new OuterInheritedSecondClass();
```

```
        secondInner.display("Second Class");
    }
}

class OuterInheritedFirstClass {
    class InheritedClass {
        InheritedClass(String name) {
            System.out.printf("Hello I am first %s
\n", name);
        }
    }
}

class OuterInheritedSecondClass extends
OuterInheritedFirstClass{
    InheritedClass inner = new
InheritedClass("InheritedClass");

    void display(String name){
        System.out.printf("I am the Second
%s\n",name);
    }
}
```

Output: