ΕΡΓΑΣΤΗΡΙΟ 11_1
Inheritance

1. What will be the output of the below program?

```
class A {
    {
        System.out.println(1);
    }
}

class B extends A {
    {
        System.out.println(2);
    }
}

class C extends B {
    {
        System.out.println(3);
    }
}

public class MainClass {
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

2. What will be the output of the following program?

```
class A {
    String s = "Class A";
}

class B extends A {
    String s = "Class B";
    {
        System.out.println(super.s);
    }
}

class C extends B {
    String s = "Class C";
    {
        System.out.println(super.s);
    }
}

public class MainClass {
    public static void main(String[] args)
    {
        C c = new C();
        System.out.println(c.s);
    }
}
```

3. What will be the output of this program?

```
class A {
    static {
        System.out.println("THIRD");
    }
}

class B extends A {
    static {
        System.out.println("SECOND");
    }
}

class C extends B {
    static {
        System.out.println("FIRST");
    }
}

public class MainClass {
    public static void main(String[] args) {
        C c = new C();
    }
}
```

4. You know that compiler will keep super() calling statement implicitly as a first statement in every constructor. What happens if we write this() as a first statement in our constructor?

5. Can you find out the error in the below code?

```
public class A {
    public A() {
        super();

        this(10);
    }

    public A(int i) {
        System.out.println(i);
    }
}
```

6. **What will be the output of this program?**

```
class M {
  static {
     System.out.println('A');
  }

  {
     System.out.println('B');
  }

  public M() {
     System.out.println('C');
    }
}

class N extends M {
    static {
     System.out.println('D');
    }

    {
     System.out.println('E');
    }

    public N() {
     System.out.println('F');
    }
}

   public class MainClass {
       public static void main(String[] args) {
           N n = new N();
       }
   }
```

7. Explain what is right and what is not right from the code below. Why?

```
package other;

public class A {
	private int i;
	int j;
	protected int k;
	public int m;
}

class B extends A {
	void methodOfClassB() {
		System.out.println(i);
		System.out.println(k);
		System.out.println(m);
	}
}

class C extends B {
	void methodOfClassC() {
		System.out.println(j);
		System.out.println(k);
		System.out.println(m);

		B b = new B();
		System.out.println(b.j);
		System.out.println(b.k);
		System.out.println(b.m);
	}
}

package lab11;

import other.A;

public class D extends A {
	void methodOfClassD() {
		System.out.println(j);
		System.out.println(k);
		System.out.println(m);

		A a = new A();
		System.out.println(a.i);
		System.out.println(a.j);
		System.out.println(a.k);
		System.out.println(a.m);
	}
}

class E extends D {
	void methodOfClassE() {
		System.out.println(k);
		System.out.println(m);

		D d = new D();
```

```
            System.out.println(d.k);

            System.out.println(d.m);
        }
}
```

8. Create a class called MusicalInstruments that should contain methods, StringInstrument, WindIntrument, and PercussionIntrument, each of which should initialize a string array to hold names of instruments of the corresponding types. Create a subclass called TypeOfIntrument that contains methods menu() and show(). The menu() method should display a menu of the types of instruments available, and the show(0 method should display the names of the type of instrument chosen by the user. The base's class variables must be accessible only to the derived class.