Kuis-2 IF2210/Pemrograman Berorientasi Objek **NIM: …… Kelas : ……..**
Jumat, 13 April 2018 **Waktu: 45 menit** **Nama: ……**

Diberikan program Java di bawah ini. Untuk setiap baris kode pada main program yang diberikan nomor dalam komentar, tuliskan pada tempat yang ditentukan:

- Jika **bisa di-compile**, tuliskan output apa yang ditampilkan, jika ada. Jika tidak ada, tuliskan "NO OUTPUT".
- Jika **tidak bisa di-compile** (*compile-error*), tuliskan "ERROR" dan tuliskan penyebab mengapa *compile-error*.

```
import java.util.ArrayList;
interface Cage<E> {
    public void add(E e);
    public void Print();
}
interface Animal {  String getSound(); }
interface Lion extends Animal {  String getFood(); }
abstract class Predator {
    protected String sound;
    protected String food;
    public Predator() {  System.out.println("New Predator");  }
    public abstract void Print();
}
class BigLion extends Predator implements Lion {
    public BigLion(String s, String f) {
        System.out.println("New BigLion");
        sound = s; food = f;
    }
    public String getSound() {  return sound;  }
    public String getFood()  {  return food;   }
    public String toString() {  return ("BigLion makes sound " + this.sound + " and eats " + this.food); }
    public void Print() {  System.out.println(this.toString()); }
}
class AnimalCage<E> implements Cage<E> {
    private ArrayList<E> animallist = new ArrayList<E>();
    public void add(E e) {
        System.out.println("Add AnimalCage");
        this.animallist.add(e);
    }
    public void Print() {
        for (E e : this.animallist) {
            System.out.println(e.toString());
        }
    }
}
class TestCage {
    public static void main (String args[]) {
        Lion king = new BigLion("rrr", "beef");                  // (1)
        Animal a = king;                                         // (2)
        Predator p = new Predator();                             // (3)

        Cage<Lion> lionCage = new AnimalCage<Lion>();            // (4)
        lionCage.add(king);                                      // (5)
        lionCage.add(a);                                         // (6)

        Cage<Animal> animalCage = new AnimalCage<Animal>();
        animalCage.add(a);                                       // (7)
        animalCage.add(king);                                    // (8)
        animalCage.Print();                                      // (9)
        animalCage = lionCage;                                   // (10)
    }
}
```

(1)	(6)
(2)	(7)
(3)	(8)
(4)	(9)
(5)	(10)