



**Universidad de las Fuerzas Armadas**

**CLASS NAME :OBJECT ORIENTED  
PROGRAMMING  
LASCANO**

**TEACHER: EDISON**

**NRC: 14575**

**WORKSHOP #: 27**

**TOPIC: Overriding and Overloading**

## 1. DAVID GUSTAVO CEPEDA SALGUERO

```
...
65 animal = new Penguin(tipperLength: 0, isIntelligent: true, speciesName: "Penguin", isTropical: true, birdColor: "Black and White", spine: "Small Spi
66 animals.add(e: animal);
67 System.out.println("animal -->" + animal);
68 System.out.println("animal type" + animal.getClass());
69
70 animal = new Owl(haghtOfVision: true, favoriteFood: "Mouse", speciesName: "Owl", isTropical: true, birdColor: "Brown", spine: "Small Spine", numberOfBones: 0
71 animals.add(e: animal);
72 System.out.println("animal -->" + animal);
73 System.out.println("animal type" + animal.getClass());
74
75 animal = new Canary(haghtOfVision: true, cageType: "Metal", speciesName: "Canary", isTropical: true, birdColor: "Yellow", spine: "Small Spine", number
76 animals.add(e: animal);
77 System.out.println("animal -->" + animal);
78 System.out.println("animal type" + animal.getClass());
79
80 Salmon salmon = new Salmon(haghtOfVision: true, averageWeight: 0, habitatType: "Ocean", isSaltwater: true, speciesName: "Salmon", fishColor: "Grey", spin
81 animals.add(e: salmon);
82 System.out.println("other salmon -->" + salmon);
83
84 salmon.swim();
85 salmon.swim(id: 1);
86
87 Crow crow = new Crow(isClever: true, foodSpoken: "Nature", speciesName: "Crow", isTropical: true, birdColor: "Black", spine: "Small Spine", numberOfBones
88 animals.add(e: crow);
89 System.out.println("other crow -->" + crow);
90
91 crow.migrate();
92 crow.migrate(id: 1);
93
94 System.out.println("\n Zoo ANIMALS");
95 System.out.println(s: animals);
96 System.out.println("total animals -->" + animals.size());
97
98 }
99
}
```

```
1 package ec.edu.espe.animal.model;
2
3 import java.util.Date;
4
5 /**
6  * @author David
7  */
8
9 public class Fish extends Vertebrate {
10     private String habitatType;
11     private boolean isSaltwater;
12     private String speciesName;
13     private String fishColor;
14
15     public Fish(String habitatType, boolean isSaltwater, String speciesName, String fishColor, String spine, int numberOfBones, int id, St
16         super(spine, numberOfBones, id, scientificName, bornDate, cageId);
17         this.habitatType = habitatType;
18         this.isSaltwater = isSaltwater;
19         this.speciesName = speciesName;
20         this.fishColor = fishColor;
21     }
22
23     public void swim() {
24         System.out.println(e: "The Fish is swimming");
25     }
26
27     @Override
28     public String toString() {
29         return "Fish(" + super.toString() + " habitatType=" + habitatType + ", isSaltwater=" + isSaltwater + ", speciesName=" + speciesName
30     }
31
32     /**
33      * Return the habitatType
34      */
35     public String getHabitatType() {
36         return habitatType;
37     }
38 }
```

