# JAVA OBJECTS

**Muhammad Ali BSSE 5B Fall 2021 080**

1. Write a program to give an example for ‘this’ operator. And use the ‘this’ keyword as return statement.

```
run:
Person's name is: John
BUILD SUCCESSFUL (total time: 0 seconds)
```

2Write a program to demonstrate static variables, methods, and blocks.

```
run:
Static block executed.
Count: 1
Count: 2
BUILD SUCCESSFUL (total time: 0 secon
```

3Write a program to give an example for ‘super’ keyword.

```
run:
Animal Sound
Bark
BUILD SUCCESSFUL (total tim
```

4Write a program to create interface A in this interface we have two method meth1 and meth2. Implements this interface in another class named MyClass.

```
run:
Method meth1() implemented in MyClass
Method meth2() implemented in MyClass
BUILD SUCCESSFUL (total time: 0 seconds)
```

5Write a program to create interface named test. In this interface the member function is square. Implement this interface in arithmetic class. Create one new class called ToTestInt in this class use the object of arithmetic class.

```
run:
Square of 5 is: 25
BUILD SUCCESSFUL (total time: 0 seconds)
```

6You are required to create a hierarchy of animals that is rooted in an abstract class Animal. Several of the animal classes will implement an interface called Pet. You will experiment with variations of these animals, their methods, and polymorphism.

```
run:
Whiskers the cat is walking.
Whiskers the cat is eating.
Whiskers the cat is playing.
Bubbles the fish cannot walk.
Bubbles the fish is eating.
Bubbles the fish is playing.
The spider is walking.
The spider is eating.
BUILD SUCCESSFUL (total time: 0 seconds)
```

6.1: the **Animal** class, which is the abstract superclass of all animals.

1. Declare a protected integer attribute called **legs**, which records the number of legs for this animal.
2. Define a protected constructor that initializes the legs attribute.
3. Declare an abstract method **eat.**
4. Declare a concrete method **walk** that prints out something about how the animals walks (include the number of legs).

```
run:
This animal with 4 legs is walking.
Whiskers the cat is eating.
Whiskers the cat is playing.
This animal with 0 legs is walking.
Bubbles the fish is eating.
Bubbles the fish is playing.
This animal with 8 legs is walking.
The spider is eating.
BUILD SUCCESSFUL (total time: 0 seconds)
```

6.2:Create the **Spider** class.

1. The Spider class extends the **Animal** class.
2. Define a default constructor that calls the superclass constructor to specify that all spiders have eight legs.
3. Implement the eat method.

```
run:
This animal with 8 legs is walking.
The spider is eating.
BUILD SUCCESSFUL (total time: 0 seconds)
```

6.3: Create the **Pet** interface specified by the UML diagram.

```
interface Pet {
   String getName();
   void setName(String name);
   void play();
}
```

The Pet interface includes three methods:

getName(): Returns a string representing the name of the pet.

setName(String name): Sets the name of the pet.

play(): Represents the action of playing with the pet.

---

6.4: Create the **Cat** class that extends Animal and implements Pet.

- This class must include a String attribute to store the name of the **pet.**
- Define a constructor that takes one String parameter that specifies the cat's name. This constructor must also call the superclass constructor to specify that all cats have four legs.
- Define another constructor that takes no parameters. Have this constructor call the previous constructor (using **this** keyword) and pass an empty string as the argument.
- Implement the Pet interface methods.
- Implement the eat method.

```
run:
This animal with 4 legs is walking.
Whiskers the cat is eating.
Whiskers the cat is playing.
Name of cat1: Whiskers
Name of cat2: Mittens
This animal with 4 legs is walking.
Mittens the cat is eating.
Mittens the cat is playing.
BUILD SUCCESSFUL (total time: 0 seconds)
```

6.5: Create the **Fish** class. Override the Animal methods to specify that fish can't walk and don't have legs.

```
run:
Bubbles the fish can't walk.
Bubbles the fish is eating.
Bubbles the fish is playing.
Name of fish1: Bubbles
Name of fish2: Nemo
Nemo the fish can't walk.
Nemo the fish is eating.
Nemo the fish is playing.
BUILD SUCCESSFUL (total time: 0 seconds)
```

6.6: Create a **TestAnimals** program. Have the main method create and manipulate instances of the classes you created above. Start with:

```
Fish d = new Fish ();
Cat c = new Cat ("Fluffy");
Animal a = new Fish ();
Animal e = new Spider ();
Pet p = new Cat ();
```

```
run:
Bubbles the fish can't walk.
Bubbles the fish is eating.
Bubbles the fish is playing.
Name of fish1: Bubbles
Name of fish2: Nemo
Nemo the fish can't walk.
Nemo the fish is eating.
Nemo the fish is playing.
BUILD SUCCESSFUL (total time: 0 seconds)
```