Dr. Faheem Akhtar Rajput

Object Oriented Programming (JAVA)

Lecture 4 & 5

Programming Paradigm

- All computer programs are conceptually organized around *Code* or *Data*. **What is happening** & **Who is being affected**

- **Process Oriented Programming:** Series of linear steps (this is, Code). We can say that Code is acting on data

- **Object Oriented Programming:** Organize a Program around its data (that is, object); Data Controlling access to code

Abstraction

- Manage complexity through abstraction
  - A car is composed of 10,000 individual parts!!!
- Breaking into more manageable pieces
- OOP is powerful natural paradigm that once you have a well defined objects, you can gracefully replace or modify the existing parts.

Three OOP Principles

Encapsulation [1/2]

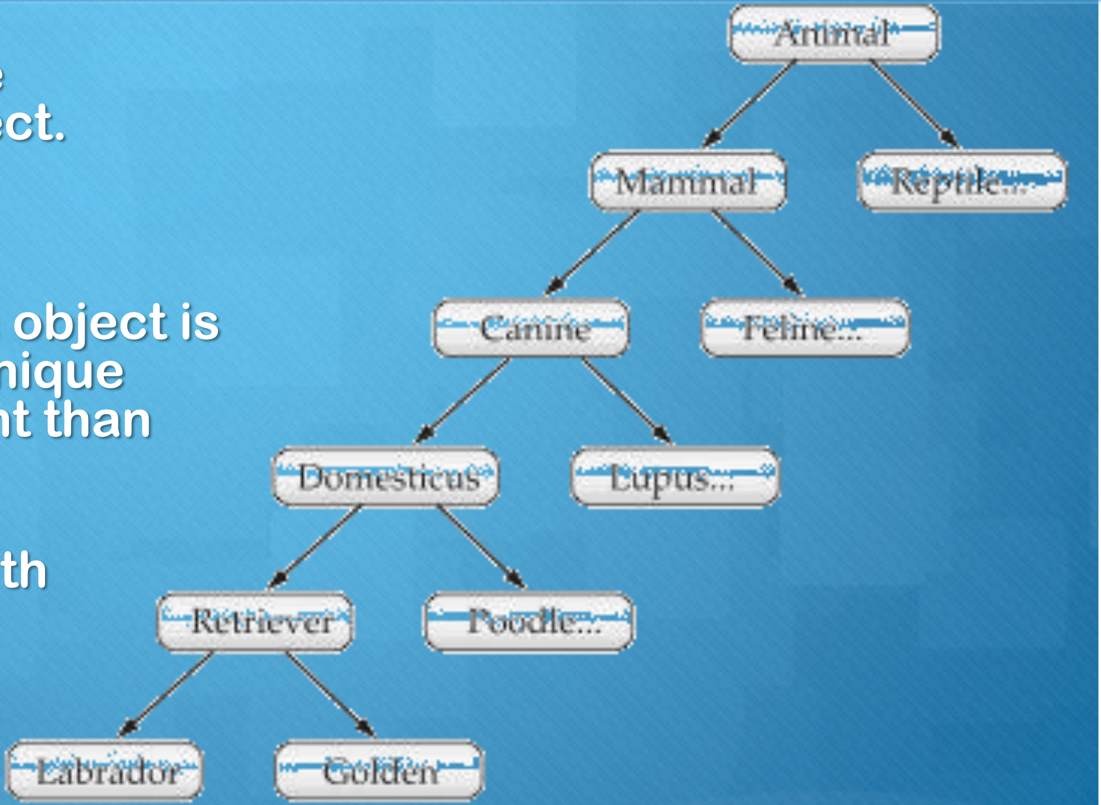
- Mechanism which binds Code & Data and keep them safe (Protective Wrapper)
- Tightly well defined interfaces can be used to access code & data
- Inside transmission doesn’t affect outside transmission
- In java basis of encapsulation is class

Encapsulation [1/2]

- Class defines structures and behaviors (Data & Code) shared by the a set of object. So each object contains structures & Behaviors
- An object is also referred as instance of a class. (Class is logical, object is physical) `Rectangle rect = new Rectangle();`
- Collectively data and code are called as members of class.
  - Member variables or variables
  - Member Methods or Methods
- Public: Everything inside the class can be accessed by outside program
- Private: Everything inside the class can only be accessed by the members of that class

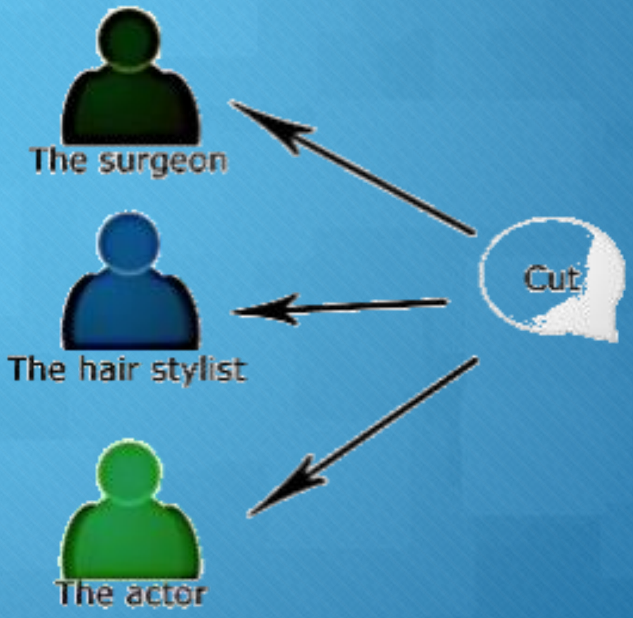
Inheritance

- One object acquires the properties of other object. Hierarchical (top down) classification
- By using Hierarchies an object is needed to define only unique characteristics (different than parents).
- Inheritance interacts with Encapsulation also



Polymorphism

- Poly (many) morph (forms)
- One interface is used for general class of actions, and the action would be determined by the nature of situation
- One interface multiple methods (A generic interface can be defined for group of related activities)
- DOG: Sense of Smell make it to bark if there’s cat and same sense make it run to bowl of milk.



Encapsulation Inheritance & Polymorphism Together

- Every java program involves encapsulation, inheritance, and polymorphism.
- All drivers rely on inheritance to drive different types (subclasses) of vehicles.
- Encapsulation is that you are dealing with a car without imagining the complexities it have.
- Polymorphism one manufacture provides several kind of vehicles some have ABS breaks some have normal braking systems. Some have manual transaction some have automatic etc..

First Java application

```
/*

This is a simple Java program.

Call this file "Example.java".

*/

class Example {

    // Your program begins with a call to main().

    public static void main(String args[]) {

         System.out.println("This is a simple Java program.");

    }

}
```

A second Short Example

```
/*
   Here is another short example.
   Call this file "Example2.java”.
*/
class Example2 {
   public static void main(String args[]) {
         int num; // this declares a variable called num
         num = 100; // this assigns num the value 100
            System.out.println("This is num: " + num);
            num = num * 2;
            System.out.print("The value of num * 2 is ");
            System.out.println(num);
     }
}
```

Two Control Statements

- Chapter 5 is dedicated for detailed discussion about controls

If statement:

```
if(condition) statement;
```

- Condition is a Boolean expression, if true then executes the statement and if false then skip the statement.

For loop:

```
for (initialization; condition; iteration) statement;
```

- Initializer sets starting of the loop
- Condition tests the control variables of loop
- Iteration gives increment or decrement to the loop.

Using Block Code

- 2 or more statements grouped together is called as *block of code* or *code block*

```
if(x < y)   { // begin a block
    x = y;
    y = 0;
} // end of block
```

- Can be targeted for any kind of block; like *if* statement or *for* statement

Questions?

Sukkur IBA

Thank you