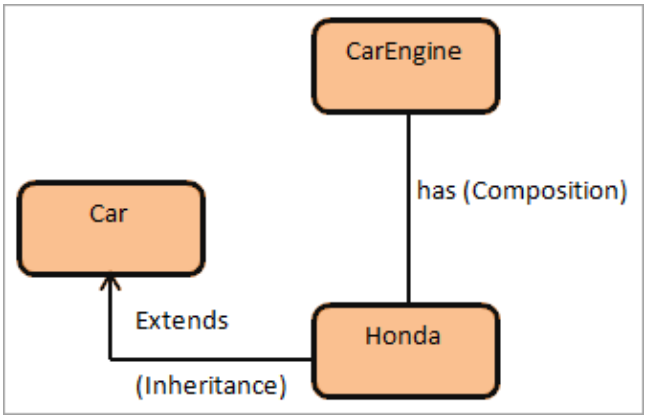
National University of Computer & Emerging Sciences,
Karachi
Computer Science Department
Spring 2023, Lab Manual – 09

Course Code: CL-1004	Course : Object Oriented Programming Lab
Instructor(s) :	Abeer Gauher, Hajra Ahmed, Shafique Rehman

LAB - 9

Containership & Nested Classes

Containership (Has a Relationship)




```
class CarEngine {
    public void startEngine(){
        System.out.println("Car Engine Started.");
    }
    public void stopEngine(){
        System.out.println("Car Engine Stopped.");
    }
}

class Car {
    private String color;
    private int max_Speed;
    public void carDetails(){
        System.out.println("Car Color= "+color + "; Max Speed= " + max_Speed);
    }
    //set car color
    public void setColor(String color) {
        this.color = color;
    }
    //set car max_Speed
    public void setMaxSpeed(int max_Speed) {
        this.max_Speed = max_Speed;
    }
}

class Honda extends Car{
    public void HondaStart(){
        CarEngine Honda_Engine = new CarEngine();   //composition
        Honda_Engine.startEngine();
        }
}
```

```
class Main {
    public static void main(String[] args) {
        Honda HondaCity = new Honda();
        HondaCity.setColor("Silver");
        HondaCity.setMaxSpeed(180);
        HondaCity.carDetails();
        HondaCity.HondaStart();
    }
}
```

Output:

```
Car Color= Silver; Max Speed= 180Car Engine Started.
```

Nested Classes

Java inner class or nested class is a class that is declared inside the class or interface.

It can access all the members of the outer class, including private data members and methods.

Syntax of Inner class

```
class Java_Outer_class{
 //code
 class Java_Inner_class{
  //code
 }
}
```

Need of Java Inner class

Sometimes users need to program a class in such a way so that no other class can access it. Therefore, it would be better if you include it within other classes.

If all the class objects are a part of the outer object then it is easier to nest that class inside the outer class. That way all the outer class can access all the objects of the inner class.

Types of Nested classes

There are two types of nested classes non-static and static nested classes. The non-static nested classes are also known as inner classes.

- Member inner class
- Anonymous inner class
- Local inner class

Java Member Inner class

A class that is created inside a class but outside a method is called member inner class. It is also known as a regular inner class. It can be declared with access modifiers like public, default, private, and protected.

Example:
In this example, we are creating a msg() method in the member inner class that is accessing the private data member of the outer class.

```
class JavaOuterClass
{
  // private variable of the outer class
  private int value = 30;
  // inner class
  class JavaInnerClass
  {
    // public variable of the inner class
    public int getValue()
    {
      System.out.println("This is the getValue method of the inner class:");
      return value;
    }
  } //inner class end here
  public static void main(String args[])
  {
    //Creating object of outer class
    JavaOuterClass outer = new JavaOuterClass();
    // Creating object of inner class
    JavaOuterClass.JavaInnerClass inner = outer.new JavaInnerClass();
    System.out.println("Value:" + inner.getValue());
  }
}
```

Output:

```
This is the getValue method of the inner class:
Value:30
```

How to instantiate Member Inner class in Java?

An object or instance of a member's inner class always exists within an object of its outer class. The new operator is used to create the object of member inner class with slightly different syntax.The general form of syntax to create an object of the member inner class is as follows:

Syntax:

OuterClassReference.new MemberInnerClassConstructor();

Example:

obj.new Inner();

Here, OuterClassReference is the reference of the outer class followed by a dot which is followed by the new operator.

Java Local inner class

A class i.e., created inside a method, is called local inner class in java. Local Inner Classes are the inner classes that are defined inside a block. Generally, this block is a method body. Sometimes this block can be a for loop, or an if clause. Local Inner classes are not a member of any enclosing classes. They belong to the block they are defined within, due to which local inner classes cannot have any access modifiers associated with them. However, they can be marked as final or abstract. These classes have access to the fields of the class enclosing it.

If you want to invoke the methods of the local inner class, you must instantiate this class inside the method.

Java local inner class example

```
public class localInner1{
 private int data=30;//instance variable
 void display(){
  class Local{
   void msg(){System.out.println(data);}
  }
  Local l=new Local();
  l.msg();
 }
 public static void main(String args[]){
  localInner1 obj=new localInner1();
  obj.display();
 }
}
```

Output:

```
30
```

Lab Tasks

Task 1:

1. Suppose you have a class called Address that contains the following instance variables: street, city, state, and zipCode.
2. You also have a class called Employee that contains the following instance variables: firstName, lastName, and address.
3. Write a Java program that creates instances of the Address class and instances of the Employee class. Each Employee object should have a unique firstName and lastName, and a random address from the Address objects created earlier.
4. Then, write a method in the Employee class that prints out the full name and address of the employee in the following format:
5. "Full Name: [firstName] [lastName]" "Address: [street], [city], [state] [zipCode]"
6. Finally, write a method in the Address class that takes in a zip code and returns a boolean indicating whether the zip code is valid or not. You can assume that a valid zip code is between 10000 and 99999.

Task 2:

- Consider a House class that has a Room class as one of its components. The Room class has attributes such as room type, room size, and room temperature.
- Write the House class with the Room class as its component, and include appropriate getter and setter methods for the Room object.
- Create an object of the House class and assign values to its attributes, including the Room object.
- Implement a method in the House class that displays the details of the house, including the details of the room.
- Write a short code snippet to create an object of the House class and display its details.

Task 3:

Suppose you have a class called ShoppingCart that contains an instance variable called itemList, which is an ArrayList of String objects. The ShoppingCart class also has a method called addItem() that takes in a String object and adds it to the itemList.

Write a Java program that creates an instance of the ShoppingCart class and adds several items to the itemList using the addItem() method. Then, write a method in the ShoppingCart class called printItems() that prints out all the items in the itemList.

Now, suppose you want to add a feature to the ShoppingCart class that sorts the itemList in alphabetical order before printing it out. You decide to use a local inner class called ItemSorter inside the printItems() method to perform the sorting.

Write the ItemSorter class inside the printItems() method, and use it to sort the itemList before printing it out. The sorting should be case-insensitive, and the items should be printed out one per line.

Task 4:

- Create a vehicle class having attributes vehicle name, engine cc, model as data members and a display function that list all the attributes of the vehicle.
- Now create a local inner class inside the display function named as Owner that has data members as owners name, CNIC number and phone number of the owner. Write down proper setters/ getters and constructors for both the classes.
- In main method create an object of class car using anonymous inner class and within that assign the owner to the car. The anonymous inner class should override the display method to show all the details of a car i.e. vehicle name, engine cc, model, Owners name and CNIC.

Task 5:

- Create a class named as "Car" that has attributes carname and cartype. Make a parameterized constructor to set these attributes. Make a private method getCarname( ) that returns car name.
- Create a class named as "Engine" that has an attribute engine type.
- Make a set engine function that first checks if the car type is equal "4T". If the condition matches, it checks if the car name is equal "Mehran" and set the engine type to small or else set the engine type to large. If not, set the engine type to "Bigger".
- The class has a method getEngineType that returns engine type.
- In the main program, create objects of the outer class as well as for the inner class. Call the functions as appropriate.