Lab - 08
Packages & Interfaces

Objectives:

Understanding the concept packages & interfaces of Java.

Theory:

Packages

Packages are containers for classes that are used to keep the class name space compartmentalized. For example, a package allows you to create a class named List, which you can store in your own package without concern that it will collide with some other class named List stored elsewhere. Packages are stored in a hierarchical manner and are explicitly imported into new class definitions.

This is the general form of the package statement:

packagepkg;

This is the general form of the import statement:

import pkg1[.pkg2].(classname|*);

Interface

Using interface, you can specify a set of methods that can be implemented by one or more classes. The interface, itself, does not actually define any implementation. Although they are similar to abstract classes, interfaces have an additional capability: A class can implement more than one interface. By contrast, a class can only inherit a single superclass (abstract or otherwise).

Lab Task:

```
// A simple package
Package MyPack;
class Balance {
String name;
Double bal;
Balance(String n, double b) {
name = n;
bal = b;
}
void show() {
if(bal<0)System.out.print("--> ");
System.out.println(name + ": $" + bal);
} }
Class AccountBalance {
public static void main(String args[]) {
Balance current[] = new Balance[3];
current[0] = new Balance("K. J. Fielding", 123.23);
current[1] = new Balance("Will Tell", 157.02);
current[2] = new Balance("Tom Jackson", -12.33);
for(inti=0; i<3; i++) current[i].show();
}
}
```

Call this file AccountBalance.java and put it in a directory called MyPack. Next, compile the file. Make sure that the resulting .class file is also in the MyPack directory. Then, try executing the AccountBalance class, using the following command line:

```
Java MyPack.AccountBalance
```

Remember, you will need to be in the directory above MyPack when you execute this command.(Alternatively, you can use one of the other two options described in the preceding section to specify the path MyPack.)
As explained, AccountBalance is now part of the package MyPack. This means that it cannot be executed by itself. That is, you cannot use this command line:

```
Java AccountBalance
```

AccountBalance must be qualified with its package name.

Interface implementation

```
interface Callback {
void callback(int param);
}
class Client implements Callback {
// Implement Callback's interface
public void callback(int p) {
System.out.println("callback called with " + p);
}
}
```

```
interface Animal
{
public void move();
}

interface Predator
{
public void hunt();
}

class Wolf implements Animal, Predator
{
private int length;
public void move()
{
   System.out.println("Wolf is moving!");
}
public void hunt()
{
   System.out.println("Wolf is hunting!");
}
}
class Fox implements Animal, Predator
{
private String fur;
public void move()
{
   System.out.println("Fox is moving!");
}
public void hunt()
{
   System.out.println("Fox is hunting!");
}
}
```

```
import java.util.ArrayList;
class AnimalPredators
{
         public static void main(String[] args) {
            ArrayList<Animal> animals=new ArrayList<Animal>();
            animals.add(new Wolf());
            animals.add(new Fox());
            animals.add(new Fox());
            for(Animal animal: animals)
            animal.move();
            ArrayList<Predator> predators=new ArrayList<Predator>();
            predators.add(new Wolf());
            predators.add(new Fox());
            predators.add(new Fox());
            for(int i=0; i<predators.size(); i++)
            predators.get(i).hunt();

}
```

Lab Assignment:

1. Design an interface Crawlable with method crawl. Create another interface Moveable with method move. Then Design a class Animal and implement both interfaces to show multiple inheritance.

2. Design a class that implements the CharSequence interface found in the java.lang package. Your implementation should return the string backwards. Select one of the sentences from this book to use as the data. Write a small main method to test your class; make sure to call all four methods.

Conclusion:

____

____

____

____

____