ASSINGMENT NO.	11
TITLE	Interfaces and packages
PROBLEM STATEMENT /DEFINITION	Write a Java program which will demonstrate a concept of Interfaces and packages: In this assignment design and use of customized interfaces and packages for a specific application are expected. Example: **Write a java program in which interface CommonList is define in package common with methods like add, remove, size, isempty, display. A class SLL in define in package specificList implements CommonList. Implements all methods given in an interface CommonList in SLL class.**
OBJECTIVE	• To understand and use of interfaces • To understand and use of packages
OUTCOME	• To be able to implement interfaces in java • To be able to use packages for specific application
S/W PACKAGES AND **HARDWARE APPARATUS USED**	• Operating Systems (64-Bit)64-BIT Fedora 17 or latest 64-BIT Update of Equivalent Open source OS or latest 64-BIT Version • Programming Tools (64-Bit) Latest Open source update of Eclipse Programming frame work, GTK+ • Programming language Java. • Editors like gedit, vi editor, etc.
REFERENCES	• The complete reference java seventh edition by Herbert Schildt • Java Programming: 24-Hour Trainer Second Edition by Yakov Fain
STEPS	Refer to steps below
INSTRUCTIONS FOR **WRITING JOURNAL**	1. Date 2. Assignment no. 3. Problem definition

	4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites:

- Knowledge of object oriented concepts.
- Basic knowledge of Java programming.

Concepts related Theory:

Interfaces:

An interface in java is a blueprint of a class. It has static constants and abstract methods. The interface in java is **a** mechanism to achieve abstraction. There can be only abstract methods in the java interface not method body. An interface declares (describes) methods but does not supply bodies for them. It is used to achieve abstraction and multiple inheritance in Java. Java Interface also represents IS-A relationship. It cannot be instantiated just like abstract class.

Example:

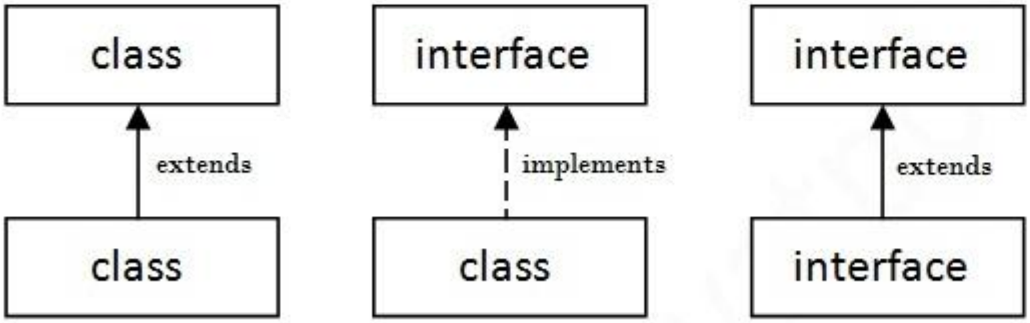
```
Interface KeyListener {
    public void keyPressed(KeyEvent e);
    public void keyReleased(KeyEvent e);
    public void keyTyped(KeyEvent e);
}
```

There are mainly three reasons to use interface. They are given below.

- It is used to achieve abstraction.
- By interface, we can support the functionality of multiple inheritance.
- It can be used to achieve loose coupling.

An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface. Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types. Method bodies exist only for default methods and static methods. Writing an interface is similar to writing a class. But a class describes the attributes and behaviors of an object. And an interface contains behaviors that a class implements. Unless the class that implements the interface is abstract, all the methods of the interface need to be defined in the class.

Understanding relationship between classes and interfaces:



Interfaces have the following properties −

- An interface is implicitly abstract. You do not need to use the abstract keyword while declaring an interface.
- Each method in an interface is also implicitly abstract, so the abstract keyword is not needed.
- Methods in an interface are implicitly public.
- You cannot instantiate an interface
- An interface is like a *very* abstract class—*none* of its methods are defined
- An interface may also contain constants (final variables)

Packages:

A java package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Packages have following properties:

- They are containers for classes/interfaces to avoid name collision
- It stored in hierarchical manner and explicitly imported into new class using import statement.
- It provide both naming and visibility control mechanism (Access Protection)
- Using it classes/interfaces can easily maintained

A Package can be defined as a grouping of related types (classes, interfaces, enumerations and annotations) providing access protection and namespace management. Some of the existing packages in Java are java.lang that bundles the fundamental classes java.io that classes for input, output functions are bundled in this package. Programmers can define their own packages to bundle group of classes/interfaces, etc. It is a good practice to group related classes implemented by you so that a programmer can easily determine that the classes, interfaces, enumerations, and annotations are related. Since the package creates a new namespace there won't be any name conflicts with names in other packages. Using packages, it is easier to provide access control and it is also easier to locate the related classes.

Algorithm:

Simple interface and package example:

```
package pr;
interface printable
{void print();  }
Package aclass;
Import pr.*;
class A implements printable{
public void print(){System.out.println("Hello");}
public static void main(String args[]){
A obj = new A();
obj.print();  }  }
```

Example of multiple inheritance by using Interface:

```
Package pr;
interface Printable{
void print();  }
interface Showable{
void show();  }
```

A.java

```
Package aclass;

Import pr.*;

class A implements Printable,Showable{

public void print(){System.out.println("Hello");}

public void show(){System.out.println("Welcome");}

public static void main(String args[]){

A obj = new A();

obj.print();

obj.show();  }}
```

Conclusion:

Students are able to implement and use concepts of interface and packages in java

Review Questions:

1. What is mean by interface and package?
2. How to use interface with respect to multiple inheritance?
3. List any five common packages used in java?
4. What are the advantages of packages?
5. What are the properties of interface?