Essentials of Information Technology

PC-CS-305

Introduction to Packages

Objectives

In this chapter, we will

- Introduce Packages
- Discuss the location and naming of packages

gauravgambhir.cse@piet.co.in

Packages in Java

- What is a Package?
- Why do we use Packages?
- Where are Packages stored?
- How are Packages located?

gauravgambhir.cse@piet.co.in

Packages

- Packages are used to group classes in a meaningful way
- There are four main reasons why packages are used:
  - To avoid name conflicts (java.util.List & java.awt.List)
  - To simplify locating classes

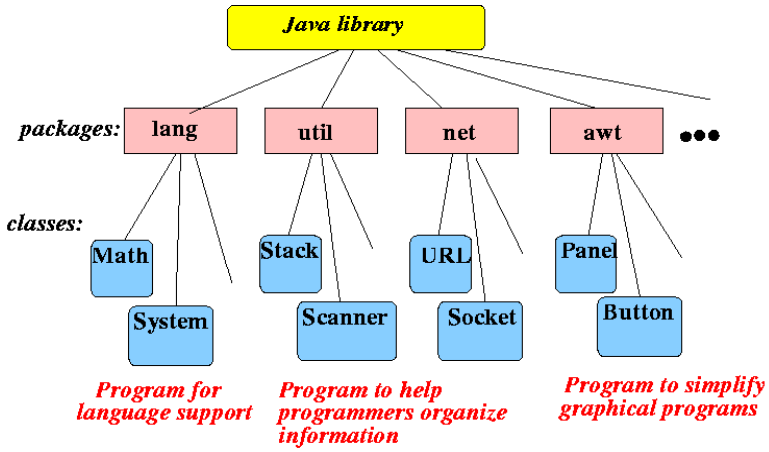
    // classes with different functionalities in different package(Slide 5)
  - To distribute software conveniently (Slide 6)
  - To control access to protected members of classes
- In a Java source file , the package that the class belong to is specified with the package keyword

  package java.awt.event;

  This keyword is usually the first keyword in the source file

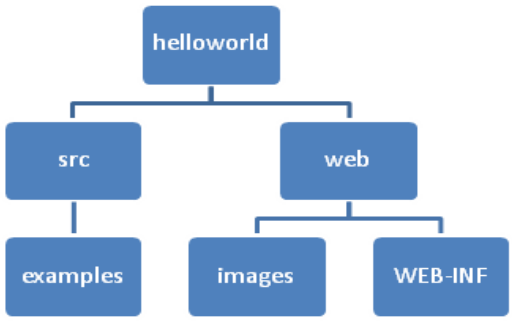
gauravgambhir.cse@piet.co.in

Java Package Hierarchy



gauravgambhir.cse@piet.co.in

Package Hierarchy Example



gauravgambhir.cse@piet.co.in

Import Statement

- The import statement is used to provide access to classes in a package:
  - import java.awt.event.*;
  - import java.util.*;
  - import javax.swing.JOptionPane;

  The first two declarations import all the classes from java.awt.event and java.util package, while last declaration import only JOptionPane from javax.swing package.

  Example : *ScannerTest.java , A.java ,B.java ,C.java*

gauravgambhir.cse@piet.co.in

Import Statement

```
import java.util.Scanner;

public class ScannerTest{
  public static void main(String arg[]){

    // scanner gets its input from the console.
    Scanner scanner = new Scanner(System.in);
    String name = "";

    // Get the user's name.
    System.out.print("Please enter your name,");
    name = scanner.next();
    System.out.println();

    // Print their name in a a message.
    System.out.println("Welcome, " + name + " to  Javaland!");
  }
}
```

```
ogy\Slides\wk11 Introduction to Packages>java ScannerTest
Please enter your name,Amit

Welcome, Amit to  Javaland!
```

Use of Import and Packages

```
package com;
public class  A
{
    public void abc()
    {
        System.out.println("I am in A");
    }

}

package example;
public class B
{
    public void def()
    {
        System.out.println("I am in B");
    }
}
```

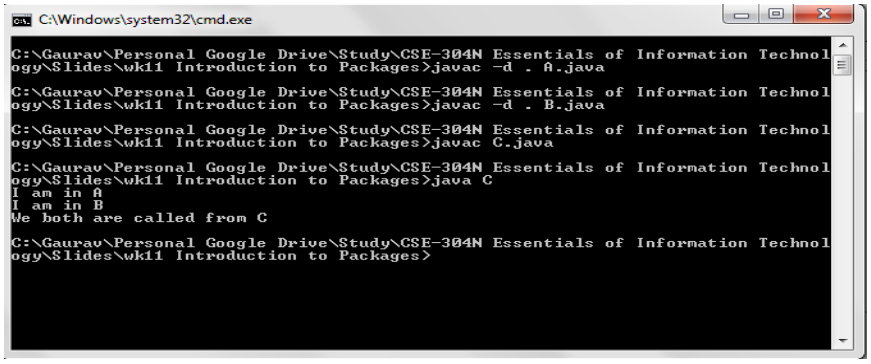
gauravgambhir.cse@piet.co.in

Use of Import and Packages

```
import com.A;
import example.B;
public class C
{
    public static void main(String[] args)
    {
        A a = new A();
        a.abc();
        B b = new B();
        b.def();
        System.out.println("We both are called from C");
    }
}
```

gauravgambhir.cse@piet.co.in

Use of Import and Packages



gauravgambhir.cse@piet.co.in

Packages and Directories

- In Java, a package is a directory that contains the bytecode of the classes.
- To avoid naming conflicts packages are given names of the domain name of the company in reverse Ex : com.microsoft, com.infosys etc.
- When a package name is not specified , a class is into the default package (the current working directory) and the package itself is given no name.
- Multiple files can specify the same package name.

gauravgambhir.cse@piet.co.in

Packages and Directories

- Always compile *all* files in a package at the same time.
  - javac *.java

gauravgambhir.cse@piet.co.in

Package Hierarchy

- Packages can be hierarchical; that is packages within packages
  - This is reflected in the directory structure
- For example I may wish to store the Card class in a package called cardgames that is in a package called mypackages in a package called pete
  - pete.mypackages.cardgames.Card
- The class would be held in a directory structure such as:
  - C:\pete\mypackages\cardgames

```
package pete.mypackages.cardgames;
public class Card
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

gauravgambhir.cse@piet.co.in

Adding Classes to a Package

- The keyword package is used to indicate that a class belongs to a certain package
- For example: *Card.java*

```
package pete.mypackages.cardgames
public class Card
{
    ....
```

- The keyword package must be the first statement in a file
  - Apart from blank lines or comments
- The package should be stored in an appropriate directory
- Other classes can be added to the same package
  - One public class per file

gauravgambhir.cse@piet.co.in

Naming Packages

- Package name should be unique
- The convention suggested is to use internet domain names in reverse order as a package prefix
- Remember internet domain names are unique
- For example:
  - Consider the domain www.fcet.staffs.ac.uk
  - This could be used as a package prefix:
  - uk.ac.staffs.fcet.pete.mypackages.cardgames
  - Example *card1.java*
- This makes directory organisation fun
- Example : Examples/P1, Examples/P2, Examples/P3

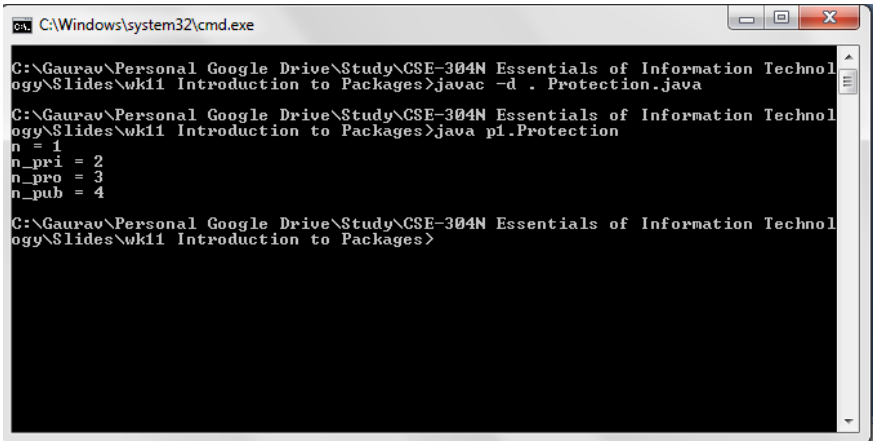
gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p1;

public class Protection {

int n = 1;
private int n_pri = 2;
protected int n_pro = 3;
public int n_pub = 4;


public void protection() {

     System.out.println("n = " + n);
     System.out.println("n_pri = " + n_pri);
     System.out.println("n_pro = " + n_pro);
     System.out.println("n_pub = " + n_pub);
}
public static void main(String args[]){
     Protection p=new Protection();
     p.protection();
}
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages



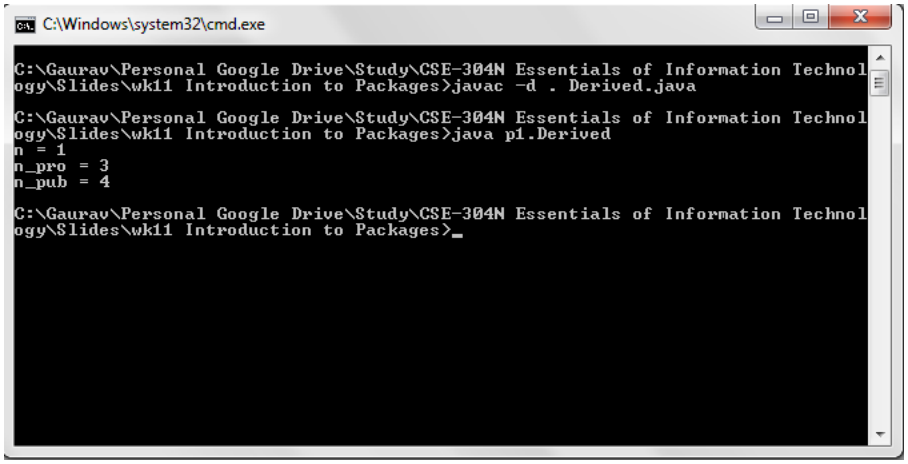
gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p1;

class Derived extends Protection {
    void derived() {


        System.out.println("n = " + n);
        // class only
        // System.out.println("n_pri = " + n_pri);
        System.out.println("n_pro = " + n_pro);
        System.out.println("n_pub = " + n_pub);
    }
    public static void main(String args[]){
        Derived d=new Derived();
        d.derived();
    }
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages



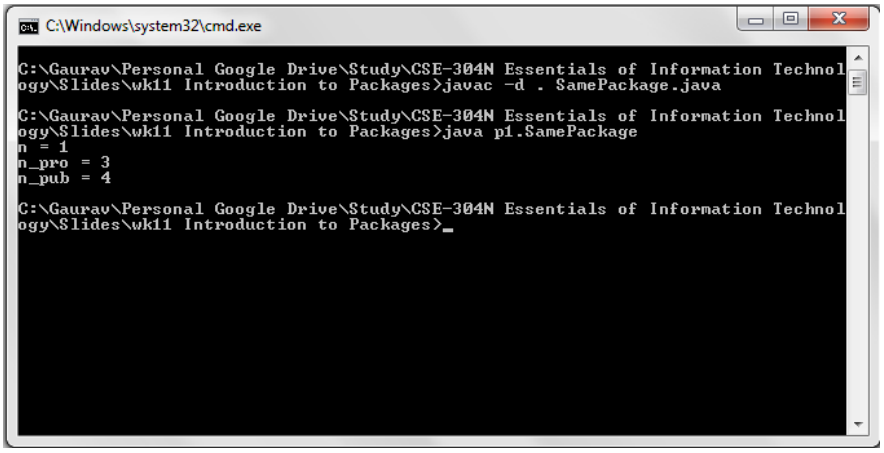
gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p1;

class SamePackage {
void samePackage() {
Protection p = new Protection();

System.out.println("n = " + p.n);
// class only
//System.out.println("n_pri = " + p.n_pri);
System.out.println("n_pro = " + p.n_pro);
System.out.println("n_pub = " + p.n_pub);
}
public static void main(String args[]){
    SamePackage p=new SamePackage();
    p.samePackage();
}
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages



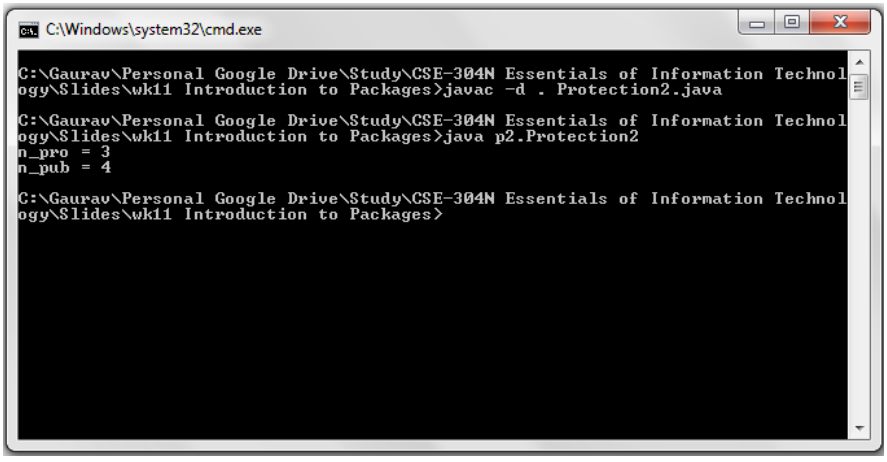
gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p2;
public class Protection2 extends p1.Protection {
public void protection2() {
// class or package only
//System.out.println("n = " + n);
// class only
// System.out.println("n_pri = " + n_pri);
System.out.println("n_pro = " + n_pro);
System.out.println("n_pub = " + n_pub);
}
public static void main(String args[]){
    Protection2 p2=new Protection2();
    p2.protection2();
}
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages



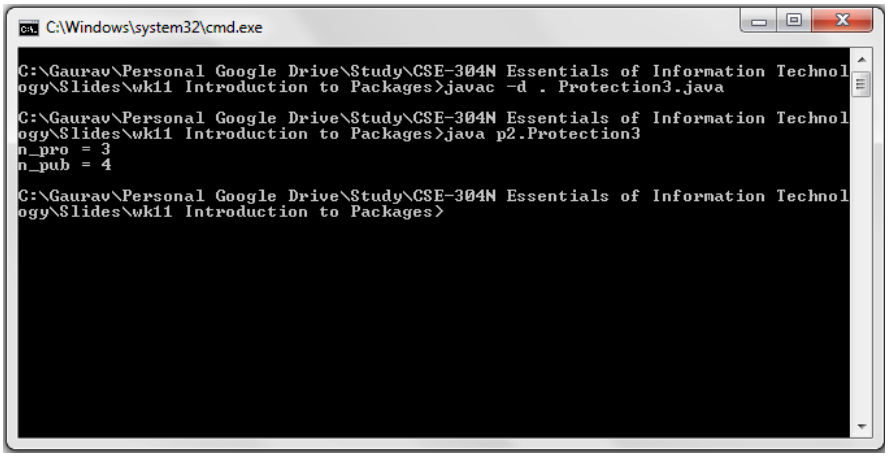
gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p2;
import p2.Protection2;
 class Protection3 extends Protection2 {
    void protection3() {
    //System.out.println("derived other package constructor");
    // class or package only
    // System.out.println("n = " + n);
    // class only
    // System.out.println("n_pri = " + n_pri);
    System.out.println("n_pro = " + n_pro);
    System.out.println("n_pub = " + n_pub);
}
public static void main(String args[]){
    //p1.Protection p = new p1.Protection();
    //System.out.println(p.n_pro);
    Protection3 p3=new Protection3();
    p3.protection3();

}
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages



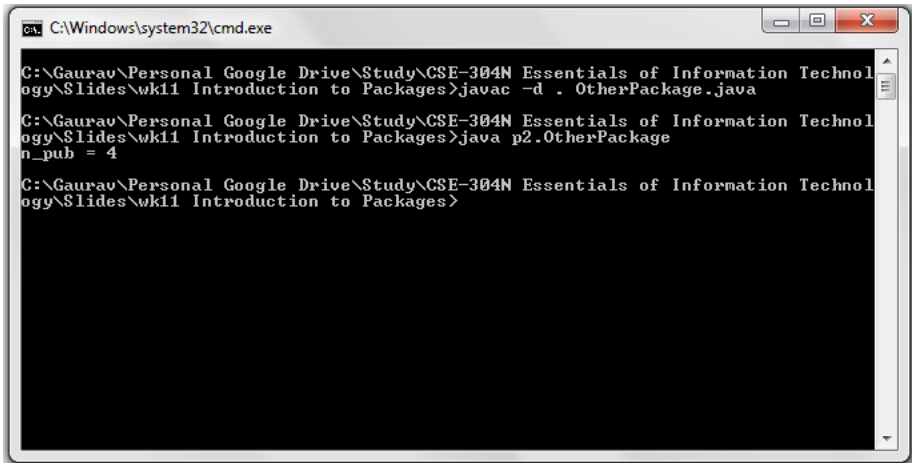
gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p2;

class OtherPackage {
    void otherPackage() {
        p1.Protection p = new p1.Protection();
            //System.out.println("other package constructor");
            // class or package only
            // System.out.println("n = " + p.n);
            // class only
            // System.out.println("n_pri = " + p.n_pri);
            // class, subclass or package only
            // System.out.println("n_pro = " + p.n_pro);
        System.out.println("n_pub = " + p.n_pub);
     }
    public static void main(String args[]){
        OtherPackage op=new OtherPackage();
        op.otherPackage();
    }

}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages



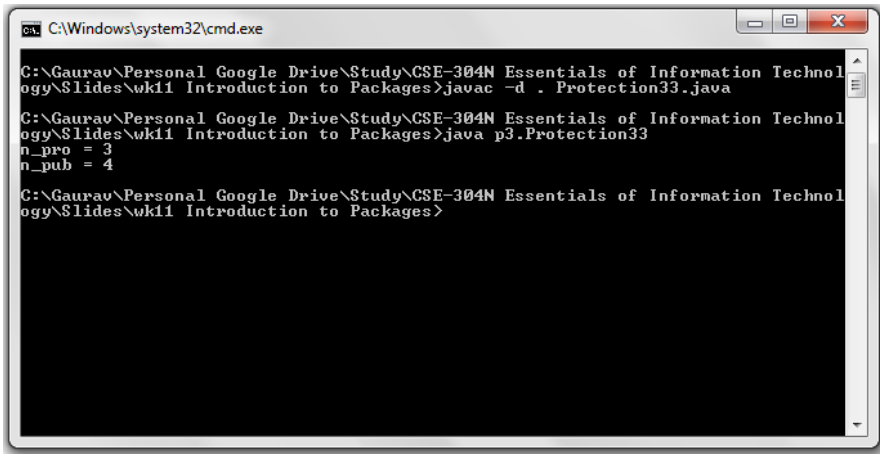
gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p3;
class Protection33 extends p2.Protection2 {
void protection33() {
    //System.out.println("derived other package constructor");
    // class or package only
    // System.out.println("n = " + n);
    // class only
    // System.out.println("n_pri = " + n_pri);
    System.out.println("n_pro = " + n_pro);
    System.out.println("n_pub = " + n_pub);
}
public static void main(String args[]){
    Protection33 p3=new Protection33();
    p3.protection33();
}
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages



gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p1;
public class A
{
    private int i=10;
    public int j=20;
    int k=30;
    protected int l=40;

    public static void main(String args[])
    {
        A a = new A();
        System.out.println(a.i); // Accessing i within same class
        System.out.println(a.j); // Accessing j within same class
        System.out.println(a.k); // Accessing k within same class
        System.out.println(a.l); // Accessing l within same class
    }
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p1;

class  B
{
    public static void main(String[] args)
    {
        A a =new A();
        //System.out.println(a.i); // Accessing i in same package but in different class.
        System.out.println(a.j); // Accessing j in same package but in different class.
        System.out.println(a.k); // Accessing k in same package but in different class.
        System.out.println(a.l); // Accessing l in same package but in different class.
    }
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p1;

class  B extends A
{
    public static void main(String[] args)
    {
        A a =new A();
        //System.out.println(a.i); // Accessing i in same package and in child class.
        System.out.println(a.j); // Accessing j in same package but in child class.
        System.out.println(a.k); // Accessing k in same package but in different class.
        System.out.println(a.l); // Accessing l in same package but in different class.
    }
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p2;
import p1.A;
public class C
{
    public static void main(String[] args)
    {
        A a =new A();
        //System.out.println(a.i);// Accessing i in different package
        System.out.println(a.j);// Accessing j in different package
        //System.out.println(a.k);// Accessing k in different package
        //System.out.println(a.l);// Accessing l in different package
    }
}
```

gauravgambhir.cse@piet.co.in

Access Modifiers & Packages

```
package p2;
import p1.A;
public class C extends p1.A
{
    public static void main(String[] args)
    {
        C a =new C();
        //System.out.println(a.i);// Accessing i in different package with parent child relation
        System.out.println(a.j);// Accessing j in different package with parent child relation
        //System.out.println(a.k);// Accessing k in different package with parent child relation
        System.out.println(a.l);// Accessing l in different package with parent child relation
    }
}
```

gauravgambhir.cse@piet.co.in

Summary

In this lecture we have:

- Introduced Packages
- Discussed the location and naming of packages

gauravgambhir.cse@piet.co.in

Question and Answer Session

Q & A

gauravgambhir.cse@piet.co.in