Introduction to Packages

Agenda

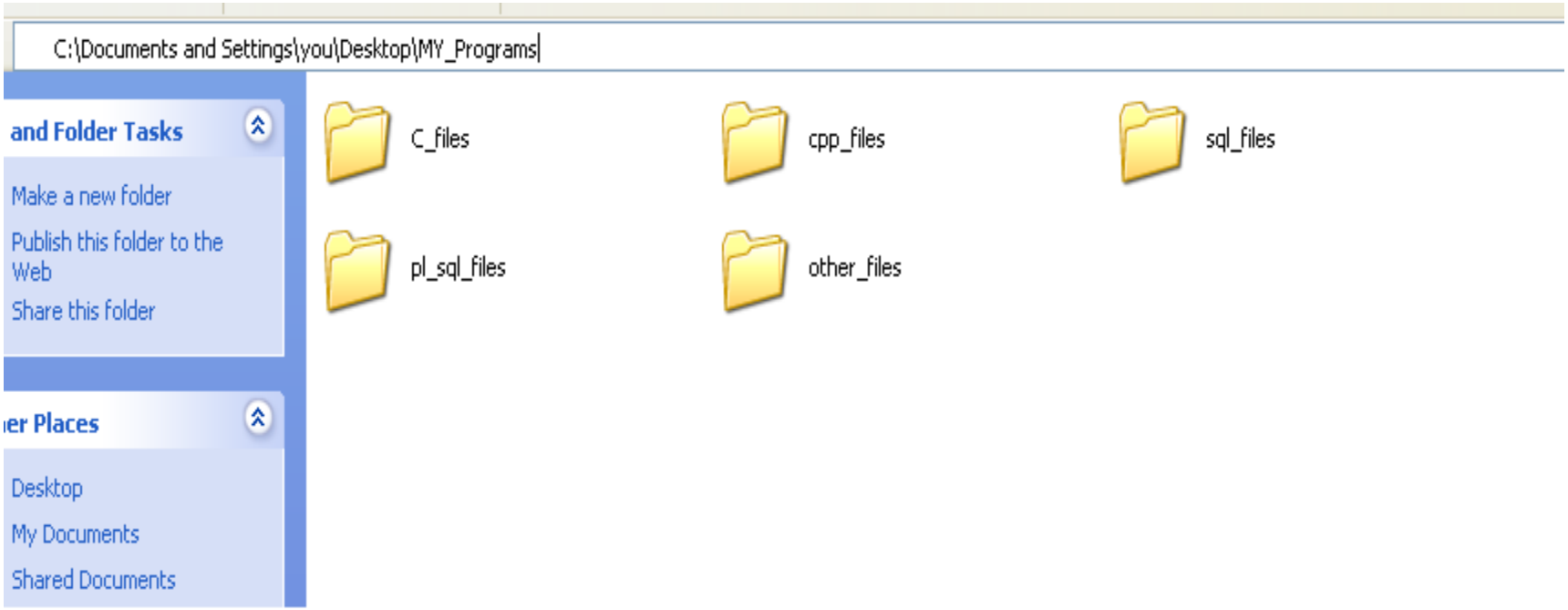
1. Introduction to Packages
2. Need for packages
3. Access protection using packages

Introduction to Packages

Package is similar to folders in your Disk

- People normally group their related data in various folders.
  - For example, a programmer may group his programs into the following folders.
  - C_programs, CPP_programs, SQL_queries, PLSQL_programs.. etc.
- We also use sub-folders for organizing our data more conveniently.
  - The advantage is we can easily locate the files if they are organized.

Package is similar to folders in your Disk (Contd.)



Just relate package concept with directories concept in your file system

Package is similar to folders in your Disk (Contd.)

- **Need for packages**
- Till now, you did not use any package since all your classes were stored in the **default** package. Imagine a situation where all the classes are stored in one package. It would lead to tremendous confusion as it may lead to classes with similar names that is not allowed by the language.

This scenario is also referred to as a namespace collision.

Organizing classes into Packages

- Packages are containers for classes and interfaces
- Classes and interfaces are grouped together in containers called **packages**
- To avoid namespace collision, we put the classes into separate containers called **packages**
- Whenever you need to access a class, you access it through its package by prefixing the class with the package name

Need for Packages

- Packages are containers used to store the classes and interfaces into manageable units of code.
- Packages also help control the accessibility of your classes. This is also called as visibility control.
- Example:

```
package MyPackage;
class MyClass {…}
class YourClass{…}
```

Access Protection using Packages

- **Packages facilitate access-control**
- Once a class is packaged, its accessibility is controlled by its package
- That is, whether other classes can access the class in the package depends on the access specifiers used in its class declaration
- There are four visibility control mechanisms packages offer:
  - private
  - no-specifier (default access)
  - protected
  - public

Packages & Access Control

Specifier	Accessibility
private	Accessible in the same class only
No-specifier (default access)	Subclasses and non-subclasses in the same package
protected	Subclasses and non-subclasses in the same package, and subclasses in other packages
public	Subclasses and non-subclasses in the same package, as well as subclasses and non-subclasses in other packages. In other words, total visibility

Packages & Access Control (Contd.)

Specifier	Accessibility
private	same class only
No-specifier (default access)	same package only
protected	same package and subclasses
public	Anywhere in the program

Real Time Scenario

In Mini's Facebook Profile page, Mini would do not wish to share some personal information like "age". But Mini would like to share "email id" with friends and Mini would like everyone to be able to access "school/university" details.

How can it be achieved?

Facebook allows users to select what level of access to be provided to their profile information.

Mini's Facebook Profile

Private Access: Age
Protected Access: Email Id
Public Access: College

Inbuilt Packages

- java language has various in-built packages.
  - java.lang, java.io, java.util, java.awt, java.applet, java.sql, javax.swing are some of the in-built packages.
  - java.lang – Basic package which is automatically imported in all programs.
    - PrintWriter, String, StringBuilder, StringBuffer, All Wrapper Classes (totally 8),
    - Thread, Runnable,
    - Throwable , Exception etc
  - java.io –Input / Output related classes are available here.
    - Scanner
    - File , FileReader , FileWriter
    - BufferedReader
    - InputStreamReader
    - IOException, FileNotFoundException etc
  - java.util – Utility classes are available here. We can use these ready-made classes.
    - Date (to work with Date) , Calender (improved one)
    - Stack (LIFO) , Queue (FIFO) , Vector , ArrayList
    - Set, HashMap etc

Inbuilt Packages(contd.).

- java.awt – Abstract Windowing Toolkit package.
  - Various classes like Button, Label, TextArea, Menu, MenuItem are available here.
- javax.swing – Swing package.
  - Various classes like JButton, JLabel, JTextArea, JMenu, JMenuItem, JTable are available here.
- java.sql – classes used for JDBC programming
  - Various classes like Connection, DriverManager, ResultSet, SQLException are available here.

Quiz

- Which is not a correct inbuilt java package?
  A) java.io
  B) java.sql
  C) java.dbms
  D) java.net

Option C is invalid package;
Others are valid java packages.

Quiz

- Which is a correct inbuilt java package?
  A) java.text
  B) java.errors
  C) java.dbms
  C) java.network

Option A is valid package;
Others are invalid java packages.

Access Specifiers in a Nutshell

Access Specifier	Private	Default	Protected	Public
Same Class	Yes	Yes	Yes	Yes
Same Package Subclass	No	Yes	Yes	Yes
Same Package Non-subclass	No	Yes	Yes	Yes
Different Package subclass	No	No	Yes	Yes
Different Package Non-Subclass	No	No	No	Yes

Summary

In this session, you were able to :

- Learn about packages

Assignment

Thank You