Object Oriented Programming

Instructor Name:

Lecture-19

Today’s Lecture

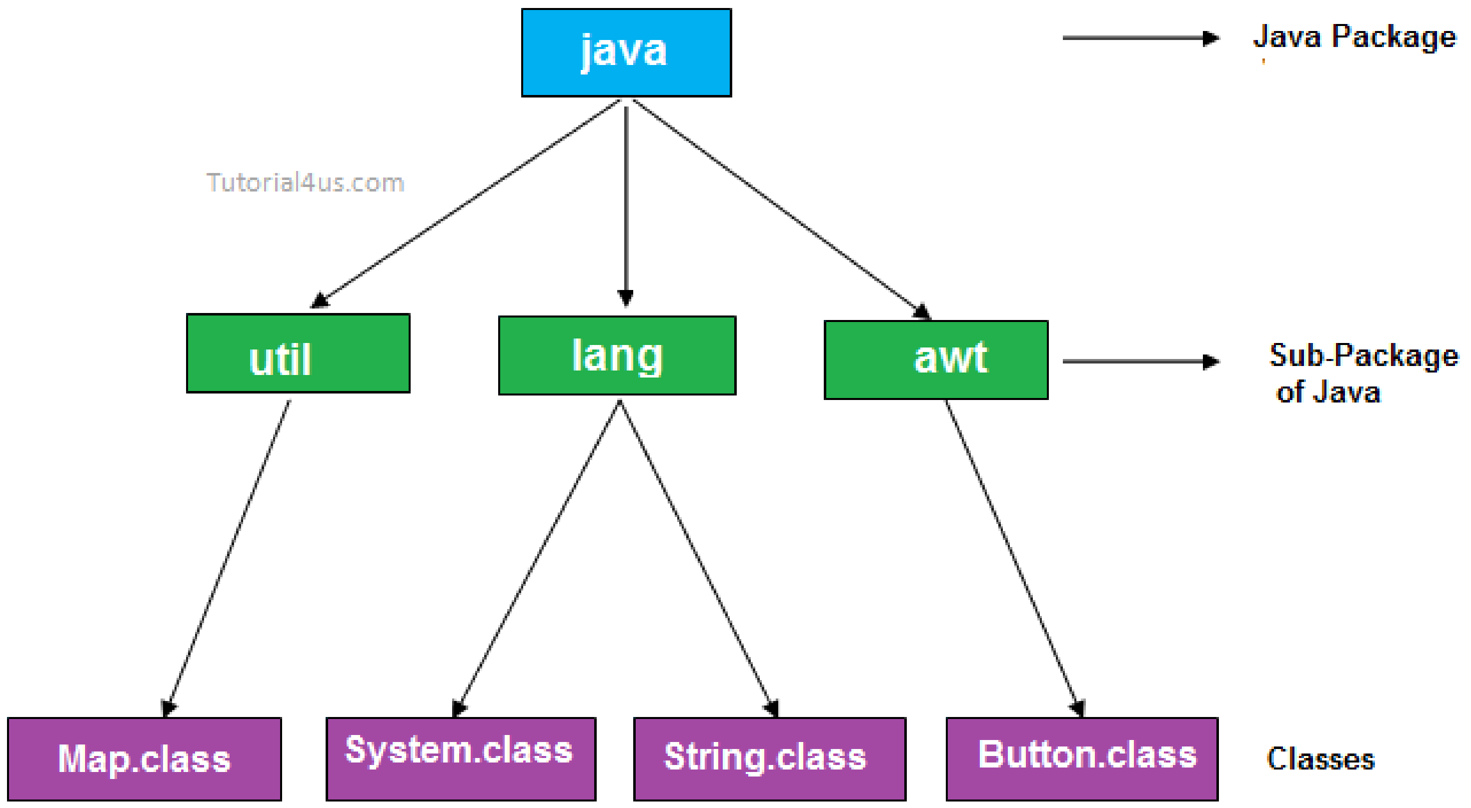
- **Packages**

Package

What is a Package?

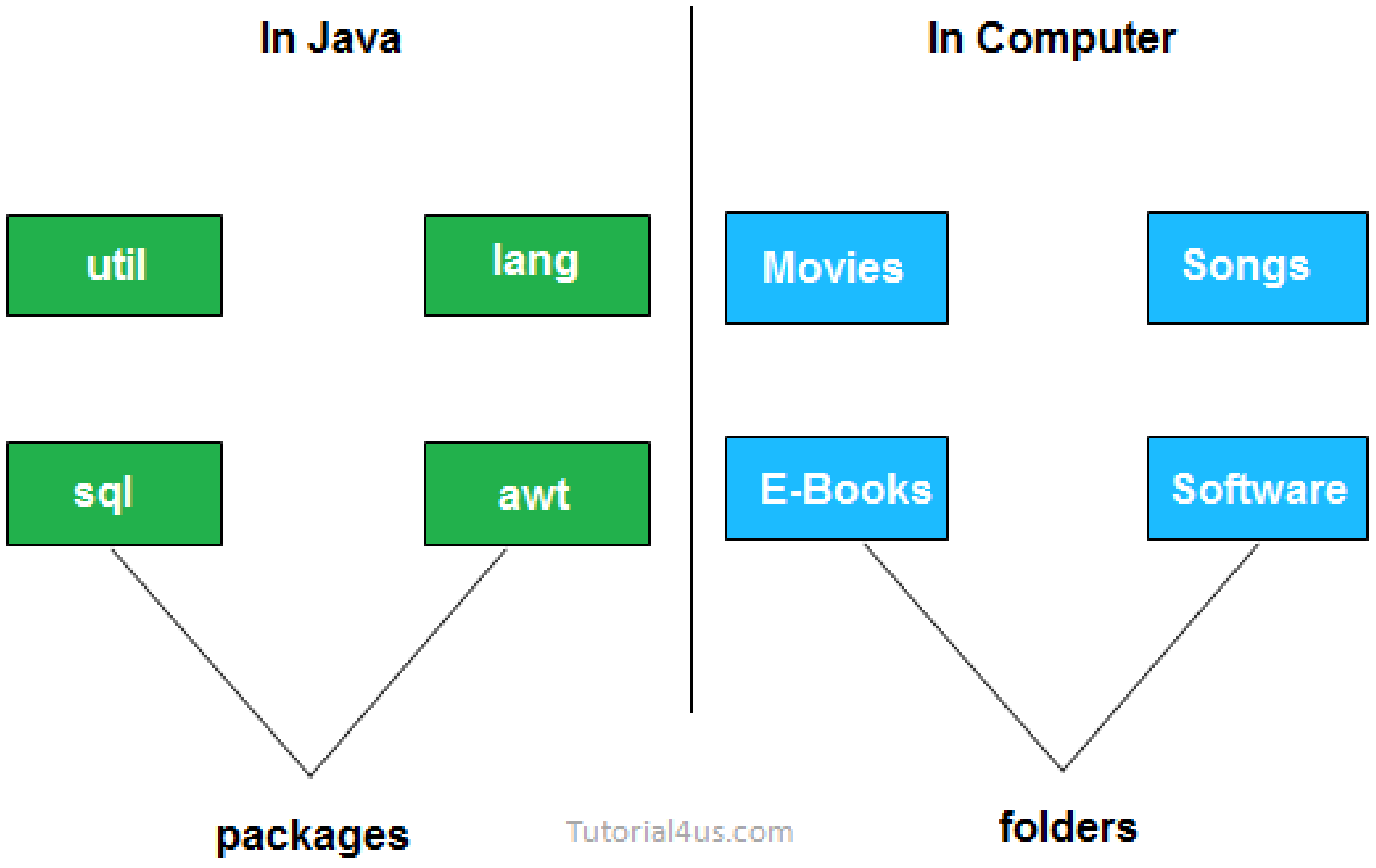
- A package is a collection of similar types of classes, interfaces and sub-packages.
- A package is a namespace that organizes a set of related classes and interfaces.
- Conceptually you can think of packages as being similar to different folders on your computer.
- Because software written in the Java programming language can be composed of hundreds or *thousands* of individual classes, it makes sense to keep things organized by placing related classes and interfaces into packages.
- The Java platform provides an enormous class library (a set of packages) suitable for use in your own applications. This library is known as the "Application Programming Interface", or "API" for short.

Package



java
Java Package
Tutorial4us.com
util
lang
awt
Sub-Package of Java
Map.class
System.class
String.class
Button.class
Classes

Package


In Java
In Computer
util
lang
Movies
Songs
sql
awt
E-Books
Software
packages
Tutorial4us.com
folders

Package

Advantages of Package

- Package is used to categorize the classes and interfaces so that they can be easily maintained
- Application development time is less, because reuse the code
- Application memory space is less (main memory)
- Application execution time is less
- Application performance is enhance (improve)
- Redundancy (repetition) of code is minimized
- Package provides access protection.
- Package removes naming collision.

Package

Types of Package

- Package are classified into two type which are given below.
  1. Predefined or built-in package
  2. User defined package
- **Predefined Packages are already designed by the Sun Microsystem and supply as a part of java API, every predefined package is collection of predefined classes, interfaces and sub-package.**
- **User Defined Packages are those which are developed by java programmer and supply as a part of their project to deal with common requirement.**

Package

Rules to Create User Defined Package

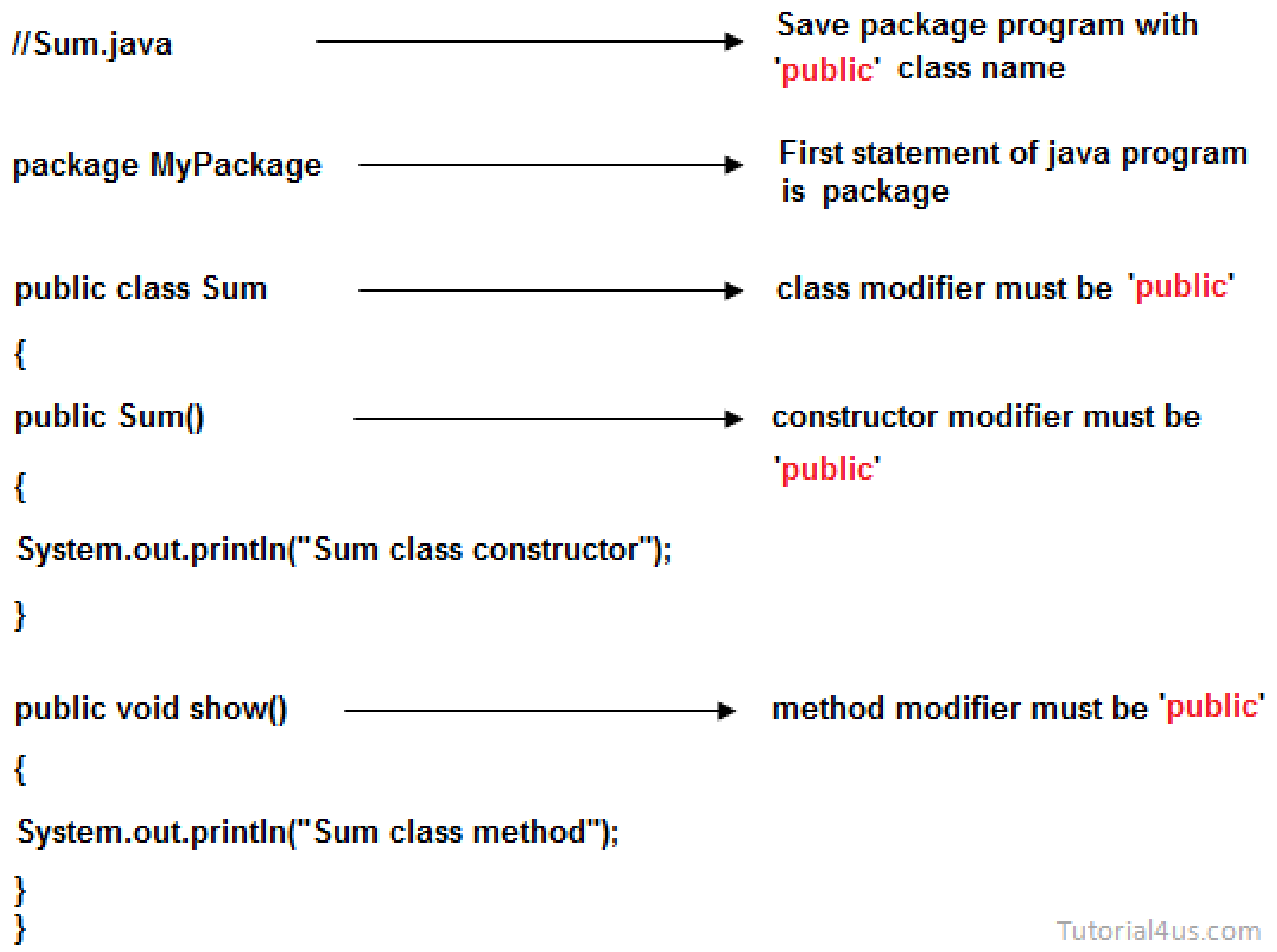
- package statement should be the first statement of any package program.
- Choose an appropriate class name or interface name and whose modifier must be public.
- Any package program can contain only one public class or only one public interface but it can contain any number of normal classes.
- Package program should not contain any main class (that means it should not contain any main())
- modifier of constructor of the class which is present in the package must be public. (This is not applicable in case of interface because interface have no constructor.)

Package

Rules to Create User Defined Package

- The modifier of method of class or interface which is present in the package must be public (This rule is optional in case of interface because interface methods by default public)
- Every package program should be save either with public class name or public Interface name

Package



```
//Sum.java

package MyPackage

public class Sum
{
public Sum()
{
System.out.println("Sum class constructor");
}

public void show()
{
System.out.println("Sum class method");
}
}
```

//Sum.java → Save package program with 'public' class name

package MyPackage → First statement of java program is package

public class Sum → class modifier must be 'public'

public Sum() → constructor modifier must be 'public'

public void show() → method modifier must be 'public'

Compiling a Package Program

- For compilation of package program first we save program with public className.java and it compile using below syntax:

```
javac -d . className.java

avac -d path className.java
```

- In above syntax "-d" is a specific tool which is tell to java compiler create a separate folder for the given package in given path.
- When we give specific path then it create a new folder at that location and when we use . (dot) then it crate a folder at current working directory.

Package

Example of Package Program

➢**Below given package example contains interface named *animals*:**

```
/* File name : Animal.java */

package animals;

interface Animal {
   public void eat();
   public void travel();
}
```

Package

Example of Package Program

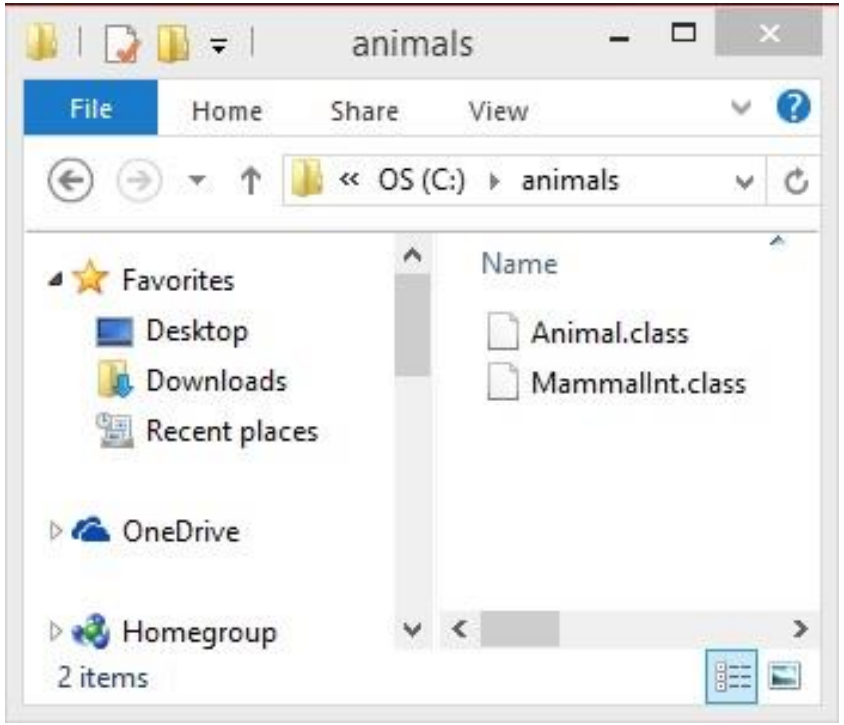
```
package animals;
/* File name : MammalInt.java */
public class MammalInt implements Animal{
    public void eat(){
            System.out.println("Mammal eats");
    }
    public void travel(){
            System.out.println("Mammal travels");
    }
    public int noOfLegs(){
            return 0;
    }
    public static void main(String args[]){
            MammalInt m = new MammalInt();
            m.eat();
            m.travel();
    }
}
```

Package

Compiling and Executing

Compile the java file as shown below

```
javac -d . Animal.java
javac -d . MammalInt.java
```



Animal.class
MammalInt.class

Execution

```
java animals.MammalInt
ammal eats
ammal travels
```

Package

Import keyword

- If a class wants to use another class in the same package, the package name does not need to be used.
- Classes in the same package find each other without any special syntax.
- Import keyword is used to include any package

Package

Example of Package Program with import

- **Package program which is save with A.java and compile by javac -d . A.java**

```
package mypack;
public class A {
  public void show() {
    System.out.println("Sum method");
  }
}
```

Package

Example of Package Program with import

- **Import above class in below program using import satatement**

```
import mypack.A;

public class Hello {
  public static void main(String arg[]) {
    A a=new A();
    a.show()
    System.out.println("show() class A");
  }
}
```

Package

Difference between Inheritance & Package

- Inheritance concept always used to reuse the feature within the program between class to class, interface to interface and interface to class but not accessing the feature across the program.
- Package concept is to reuse the feature both within the program and across the programs between class to class, interface to interface and interface to class.

Package

package keyword vs import keyword

- package keyword is always used for creating the undefined package and placing common classes and interfaces.
- import is a keyword which is used for referring or using the classes and interfaces of a specific package.

Package

Directory Structure of packages

- **Two major results occur when a class is placed in a package:**
  1. **The name of the package becomes a part of the name of the class, as we have seen in the previous code.**

```
import mypack.A;
```

  2. **The name of the package must match the directory structure where the corresponding bytecode resides.**

String Tokenizer

What is String Tokenizer?

- It is a pre defined class in java.util package can be used to split the given string into tokens (parts) based on delimiters (any special symbols or spaces).
- Suppose that we have any string like "Features of Java_Language" when we use stringTokenizer this string is split into tokens whenever spaces and special symbols present.
- After split string are :

Features

of

Java

Language

String Tokenizer

Methods of String Tokenizer

- **hasMoreTokens()**

  It is predefined method of StringTokenizer class used to check whether given StringTokenizer having any elements or not.

- **nextToken()**

  Which can be used to get the element from the StringTokenizer.

String Tokenizer

Example of StringTokenizer

```
import java.util.*;
class Stringtokenizerdemo {
  public static void main(String args[]) {
   String str="He is a gentle man";
   StringTokenizer st=new StringTokenizer(str," ");

   System.out.println("The tokens are: ");
   while(st.hasMoreTokens()) {
         String one=st.nextToken();
         System.out.println(one);
   }
  }
}
```

String Tokenizer

Example of StringTokenizer

Output

```
The tokens are:
He
is
a
gentle
man
```

R@CAP