

REC ●



welcome to my
presentation



JAYA SURIYA
20384112
COMPUTER SCIENCE

PACKAGES

JAVA PROGRAMMING

TOPIC OUTLINE



WHAT IS
PACKAGES



TYPES OF
PACKAGES



CREATE & ACCESS
A PACKAGES

INTRODUCTION

WHAT IS PACKAGES?

A java package is a group of similar classes, interface and sub-packages.

package in java can be categorized in two form, built-in (or)java API package & user-defined package.



A young man with dark purple hair is sitting on a light blue couch. He is wearing a purple hoodie with green stripes on the sleeves and light blue jeans. He has a thoughtful expression, with his right hand resting against his chin and his head tilted slightly to the side. The background behind him is a dark purple gradient.

API PACKAGES:

Two light blue paw prints are positioned on the right side of the slide, aligned vertically. They are simple, stylized representations of animal paws.

JAVA API PROVIDES A LARGE NUMBER OF CLASSES GROUPED INTO DIFFERENT PACKAGES ACCORDING TO FUNCTIONALITY . MOST OF THE TIME WE USE THE PACKAGES.

java subpackage & their classes:

- JAVA. LANG - LANGUAGE support classes. they include classes for primitive, strings, math functions, threads and exceptions.
- JAVA. UTIL - UTILITY classes such as vectors, Hash tables, Random numbers, Date, etc..
- JAVA. io - INPUT\OUTPUT support classes. They provide facilities for the input and output of data.
- JAVA. AWT - SET of classes for implementing graphical user interface. They include classes for Windows, Buttons, Lists, Menu and so on.

- JAVA.NET - IT FOR NETWORKING. They include classes for communicating with local computers as well as With INTERNET SERVERS.
- JAVA. APPLET - It is CREATING AND IMPLEMENTING APPLETS.

user-defined packages:

As the name propose, user-defined packages in java are essentially packages that are defined by the programmer. whenever we want to add a class to the package,we have to mention the packages name and the " package" Keyword at the top of the program.





SYNTAX:

package package-name;

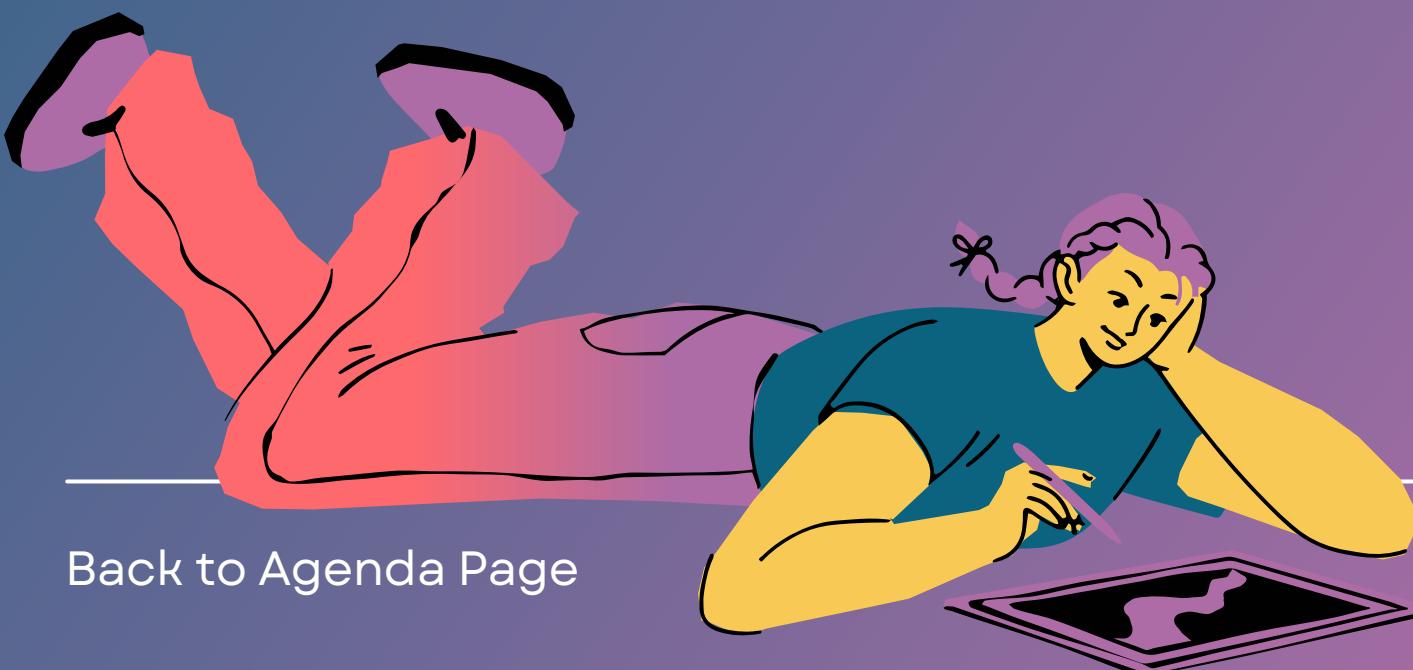
steps to create user- defined package:

step 1: creating a package in java class.
the format is very simple and easy.just write a package by following by following its name. " package example1; "

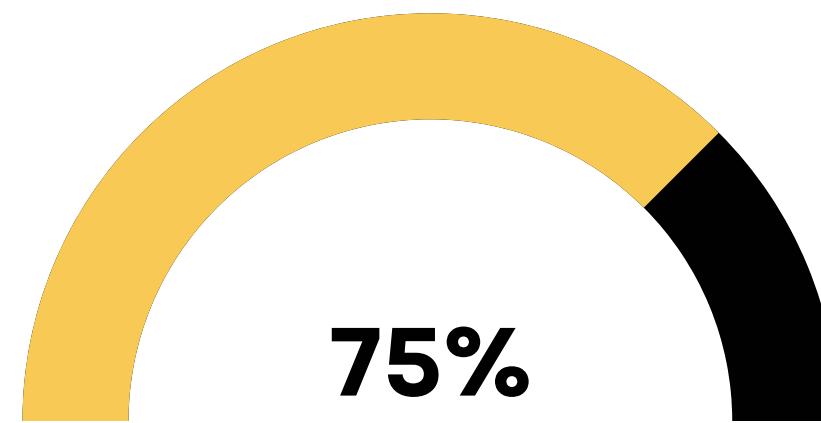
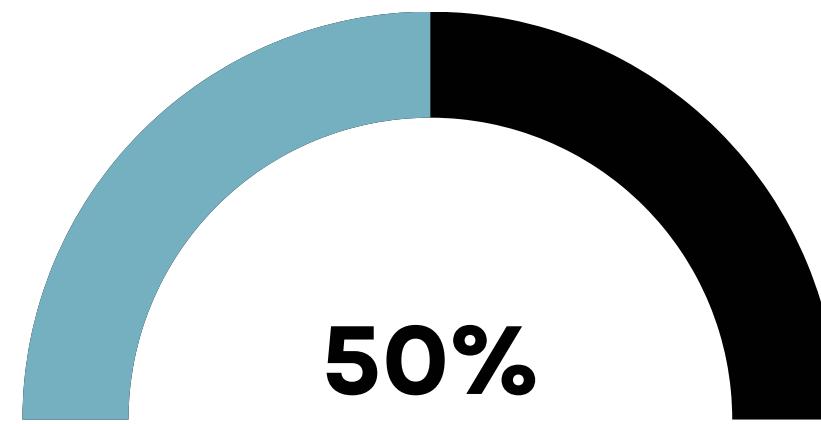
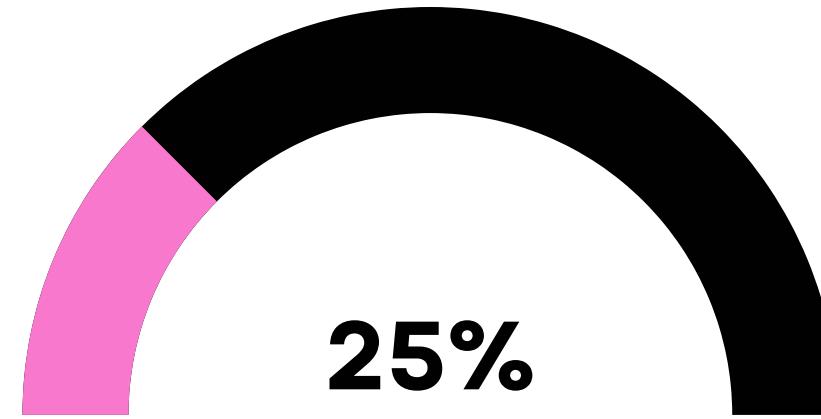
step 2: include class in java package,
but remember that class only has one package declartion.

step 3: Now the user-defined package is successfully created, we can import it other packages and use its functions it.

advantag e of java package:



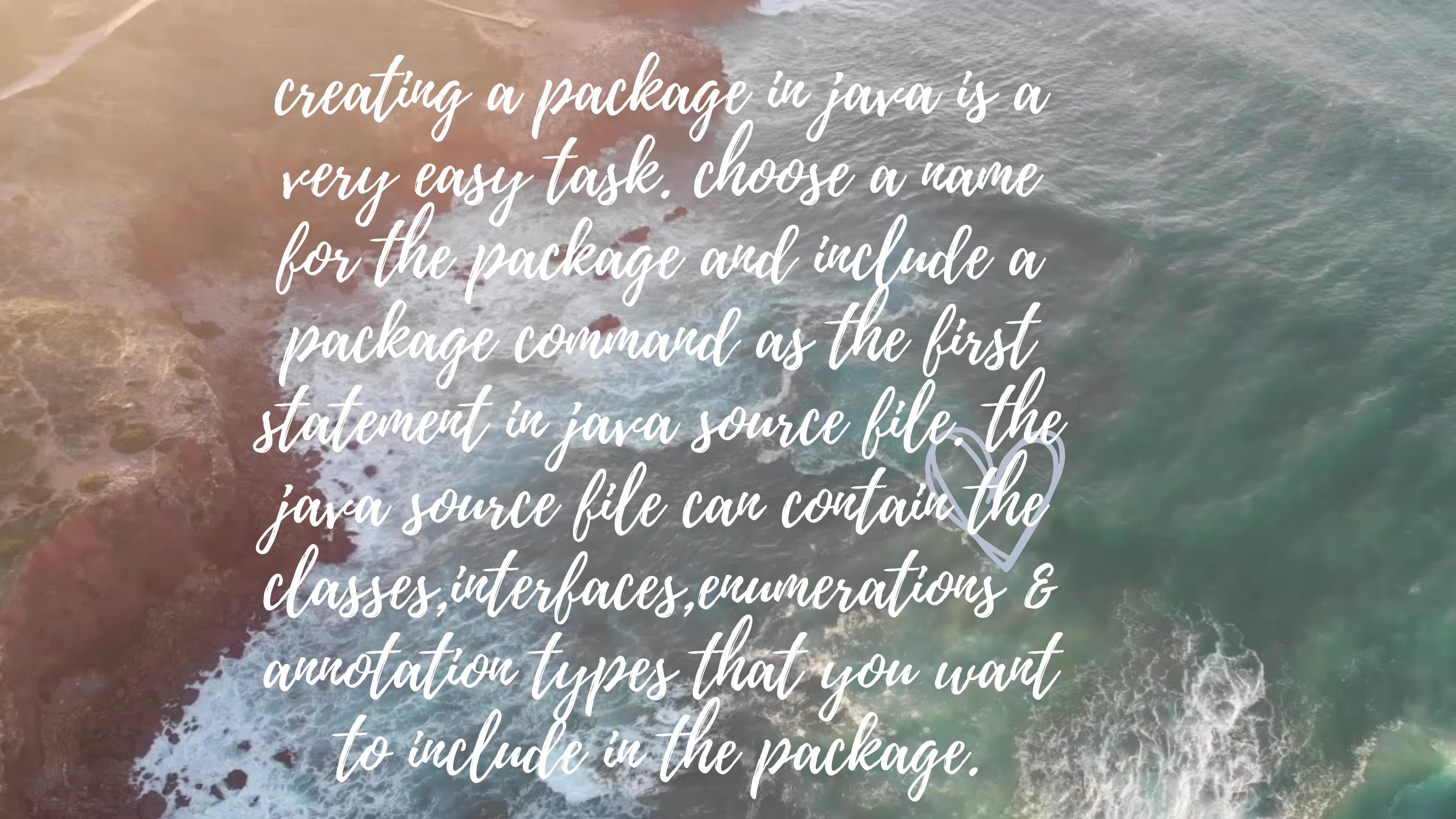
Back to Agenda Page



1) java package is used to categorize the classes and interface so that they can be easily maintained.

2) java package provides access protection.

3) java package removes naming collision.



creating a package in java is a very easy task. choose a name for the package and include a package command as the first statement in java source file. the java source file can contain the classes, interfaces, enumerations & annotation types that you want to include in the package.



HOW TO ACCESS THE PACKAGE ?

Canva

**THERE ARE THREE WAYS TO
ACCESS THE PACKAGE
FROM OUTSIDE THE
PACKAGE:**

- 1. IMPORT PACKAGE.*;**
- 2. IMPORT
PACKAGE.CLASSNAME;**
- 3. FULLY QUALIFIED NAME.**

@reallygreatsite

PACKAGENAME.::

```
//SAVE BY A.JAVA  
PACKAGE PACK;  
PUBLIC CLASS A{  
    PUBLIC VOID MSG(){SYSTEM.OUT.PRINTLN("HELLO");}  
}
```

```
//SAVE BY B.JAVA  
PACKAGE MYPACK;  
IMPORT PACK.*;
```

```
CLASS B{  
    PUBLIC STATIC VOID MAIN(STRING ARGS[]){  
        A OBJ = NEW A();  
        OBJ.MSG();  
    }  
}
```

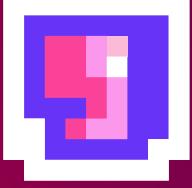
IF YOU USE PACKAGE.* THEN ALL THE CLASSES AND INTERFACES OF THIS PACKAGE WILL BE ACCESSIBLE BUT NOT SUBPACKAGES.

THE IMPORT KEYWORD IS USED TO MAKE THE CLASSES AND INTERFACE OF ANOTHER PACKAGE ACCESSIBLE TO THE CURRENT PACKAGE.



OUTPUT: HELLO

PACKAGENAME.CLASSNAME



//SAVE BY A.JAVA

```
PACKAGE PACK;  
PUBLIC CLASS A{  
    PUBLIC VOID MSG(){SYSTEM.OUT.PRINTLN("HELLO");}  
}
```

//SAVE BY B.JAVA
PACKAGE MYPACK;
IMPORT PACK.A;

```
CLASS B{  
    PUBLIC STATIC VOID MAIN(STRING ARGS[]){  
        A OBJ = NEW A();
```

```
        OBJ.MSG();  
    }  
}
```



IF YOU IMPORT
PACKAGE.CLASSNAME THEN
ONLY DECLARED CLASS OF
THIS PACKAGE WILL BE
ACCESSIBLE.

OUTPUT:HELLO

FULLY QUALIFIED NAME

```
//SAVE BY A.JAVA  
PACKAGE PACK;  
PUBLIC CLASS A{  
    PUBLIC VOID MSG(){SYSTEM.OUT.PRINTLN("HELLO");}  
}  
//SAVE BY B.JAVA  
PACKAGE MYPACK;  
CLASS B{  
    PUBLIC STATIC VOID MAIN(STRING ARGS[]){  
        PACK.A OBJ = NEW PACK.A(); //USING FULLY QUALIFIED NAME  
        OBJ.MSG();  
    }  
}
```

IF YOU USE FULLY QUALIFIED NAME THEN ONLY DECLARED CLASS OF THIS PACKAGE WILL BE ACCESSIBLE. NOW THERE IS NO NEED TO IMPORT. BUT YOU NEED TO USE FULLY QUALIFIED NAME EVERY TIME WHEN YOU ARE ACCESSING THE CLASS OR INTERFACE.

OUTPUT: HELLO



THANK YOU
ALL!! *Anne*