**Summary of Java Concepts: Packages, Access Modifiers, and Imports**

**1. Creating and Using Packages**

* **Definition**: Packages are namespaces for grouping related classes and interfaces. They:
  + Organize code logically.
  + Prevent naming conflicts.
  + Control class access.

**Creating a Package**:

java

CopyEdit

package com.example.animals;

public class Dog {

public void sound() {

System.out.println("Barking...");

}

}

**Using the Package**:

java

CopyEdit

import com.example.animals.Dog;

public class Main {

public static void main(String[] args) {

Dog dog = new Dog();

dog.sound(); // Outputs: Barking...

}

}

**Key Note**: Ensure your directory structure matches the package name (e.g., com/example/animals/ for com.example.animals).

**2. Access Modifiers**

Access modifiers determine the visibility of classes, methods, and variables.

| **Modifier** | **Class** | **Package** | **Subclass** | **World** |
| --- | --- | --- | --- | --- |
| public | Yes | Yes | Yes | Yes |
| protected | Yes | Yes | Yes | No |
| Default | Yes | Yes | No | No |
| private | Yes | No | No | No |

**Example**:

java

CopyEdit

class Animal {

public String name; // Accessible everywhere

private int age; // Accessible only in Animal

protected String color; // Accessible in package and subclasses

String species; // Default access, package-private

}

* public: Accessible everywhere.
* private: Only accessible within the defining class.
* protected: Accessible within the same package or subclasses.
* Default (no modifier): Accessible only within the same package.

**3. The import Statement**

Used to bring classes or packages into scope, avoiding fully qualified names.

**Importing a Single Class**:

java

CopyEdit

import com.example.animals.Dog;

Dog dog = new Dog(); // No need for com.example.animals.Dog

**Importing All Classes from a Package**:

java

CopyEdit

import com.example.animals.\*;

Dog dog = new Dog(); // Automatically available

**4. Static Imports**

Static imports allow direct access to static members of a class without using the class name.

**Example**:

java

CopyEdit

import static java.lang.Math.PI;

import static java.lang.Math.sqrt;

public class Main {

public static void main(String[] args) {

System.out.println("PI: " + PI); // Outputs: 3.141592653589793

System.out.println("Sqrt of 16: " + sqrt(16)); // Outputs: 4.0

}

}

**Key Takeaways**

* **Packages**: Use the package keyword to organize and manage classes.
* **Access Modifiers**: Control visibility with public, private, protected, and default (package-private).
* **Import Statements**: Simplify references to external classes or packages.
* **Static Imports**: Facilitate direct use of static members for cleaner and more concise code.