



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 5
Implement a program on Packages.
Date of Performance:
Date of Submission:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Aim: To use packages in java.

Objective: To use packages in java to use readymade classes available in them using square root method in math class.

Theory:

A java package is a group of similar types of classes, interfaces and sub-packages. Packages are used in Java in order to prevent naming conflicts, to control access, to make searching/locating and usage of classes, interfaces, enumerations and annotations easier, etc.

There are two types of packages-

1. Built-in package: The already defined package like java.io.*, java.lang.* etc are known as built-in packages.
2. User defined package: The package we create for is called user-defined package.

Programmers can define their own packages to bundle group of classes/interfaces, etc. While creating a package, the user should choose a name for the package and include a package statement along with that name at the top of every source file that contains the classes, interfaces, enumerations, and annotation types that you want to include in the package. If a package statement is not used then the class, interfaces, enumerations, and annotation types will be placed in the current default package.

Code:

// Save In A.Java In A New Folder Pack2

```
package mypack;
public class A
{
    public static void msg()
    {
        System.out.println("Welcome to Package!!");
    }
}
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

// Save in B.java

```
package mypack;
import mypack.*;
class B
{
    public static void main(String args[])
    {
        A obj=new A();
        obj.msg();
    }
}
```

Output:

```
C:\Users\Chaitanya24\Desktop\OOPJ Exp>javac ./mypack/A.java
C:\Users\Chaitanya24\Desktop\OOPJ Exp>javac B.java
C:\Users\Chaitanya24\Desktop\OOPJ Exp>java B.java
Welcome to Package!!
```

Conclusion:

Comment on the autoencoder architecture and the Image compression results.

- The autoencoder architecture and the image compression results are related to the topic of lossy image compression using deep learning. Lossy image compression is a technique that reduces the size of an image file by discarding some information that is not essential for human perception. The goal is to achieve a high compression ratio while maintaining a good visual quality.
- An autoencoder is a type of neural network that can learn to compress and reconstruct data. It consists of two parts: an encoder and a decoder. The encoder takes an input image and transforms it into a lower-dimensional representation, called the latent code or the bottleneck. The decoder takes the latent code and reconstructs the original image as closely as possible. The autoencoder is trained to minimize the reconstruction error, which is the difference between the input and the output images.
- The results from the web search show that autoencoder-based lossy image compression can achieve better performance than existing image compression standards, such as JPEG and JPEG2000, in terms of compression ratio and visual quality. For example, according to [Lossy Image Compression with Compressive Autoencoders](#), a deep convolutional autoencoder can compress an image by up to 98% with minimal distortion. The paper also claims that autoencoder-based image compression can be more adaptable to new media contents and formats, such as 360-degree images and virtual reality