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

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!!");
```

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

Conclusion:

Comment on the autoencoder architecture and the Image compression results.
Ans: Autoencoders are a type of artificial neural network used for unsupervised learning and dimensionality reduction. They consist of an encoder and a decoder, with the goal of learning a compressed representation (encoding) of the input data. The architecture of an autoencoder can be summarized as follows:

1. Encoder: The encoder takes the input data and maps it to a lower-dimensional representation, often referred to as a "bottleneck" or "latent space." The encoder network typically consists of multiple layers, reducing the dimensionality of the data in a hierarchical manner. Each layer uses activation functions (e.g., ReLU, Sigmoid) to introduce non-linearity and learn features from the data. The final layer of the encoder produces the compressed representation.

2. Decoder: The decoder takes the compressed representation from the encoder and attempts to

reconstruct the original input data. Like the encoder, it typically consists of multiple layers, expanding the dimensionality of the data back to the original input size. The activation functions used in the decoder layers are usually chosen to match the characteristics of the input data (e.g., Sigmoid for pixel values in the range [0, 1] for image data).

Autoencoders can be used for various tasks, including data compression, image denoising, anomaly detection, and feature learning.

Regarding image compression results with autoencoders, here are some key points to consider:

1. Lossy Compression**: Autoencoders typically perform lossy compression, meaning that the reconstructed data may not be an exact match to the original input. The quality of reconstruction depends on the architecture of the autoencoder, the size of the latent space, and the training data.

2. Compression Ratio: The compression ratio is determined by the size of the latent space compared to the size of the input data. A smaller latent space leads to higher compression, but it may result in a loss of important information.

3. Image Quality: The quality of compressed images can vary. Some details may be lost in the compression process, especially if the latent space is too small. The choice of loss function during training also impacts the visual quality of the reconstructed images.

Autoencoders can be used for image compression by learning a compact representation of images, but the trade-offs between compression ratio and image quality need to be carefully considered based on the specific requirements of the application. The effectiveness of an autoencoder-based image compression system can be assessed using appropriate evaluation metrics to ensure it meets the desired quality standards.