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:

```
package mypack;
public class Simple{
public static void main(String args[])
{
System.out.println("Welcome to package");
}
 }
```

OUTPUT:

```
C:\Users\yedu0\OneDrive\Documents\java>javac Simple.java

C:\Users\yedu0\OneDrive\Documents\java>java Simple.java
Welcome to package
```

Conclusion:

Comment on the autoencoder architecture and the Image compression results.

Autoencoders are a type of artificial neural network used for learning efficient representations of data, typically for dimensionality reduction, data denoising, or compression. The basic architecture of an autoencoder consists of an encoder and a decoder. The encoder compresses the input data into a lower-dimensional representation, and the decoder reconstructs the original input from this representation. Encoder: The encoder network maps the input data to a latent space representation, which is typically of lower dimensionality than the input. This compressed representation retains the most important features of the input data.

Decoder: The decoder network reconstructs the original input from the latent space representation generated by the encoder. It aims to produce an output that is as close as possible to the original input.