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:

```
1. package mypack;
class simple{
public static void main(String args[]){
System.out.println("ketan mahadik");
}
}
```

OUTPUT:

```
C:\Users\ketan\OneDrive\Desktop\java>javac -d . simple.java

C:\Users\ketan\OneDrive\Desktop\java>java mypack.simple
ketan mahadik

C:\Users\ketan\OneDrive\Desktop\java>
```

Conclusion:

Comment on the autoencoder architecture and the Image compression results.

I've explored autoencoder architectures in the context of image compression and observed the following:

1. Autoencoder Architecture:
   - Autoencoders are neural networks used for unsupervised learning and data compression.
   - They consist of an encoder and a decoder. The encoder reduces the input data into a lower-dimensional representation (encoding), and the decoder reconstructs the original data from this encoding.
   - Various architectures can be employed, including shallow autoencoders, deep autoencoders, and convolutional autoencoders, depending on the complexity and type of data.

2. Image Compression Results:
   - Autoencoders have been applied effectively to image compression tasks.
   - By training an autoencoder on a dataset of images, I've observed that it learns to represent images with a reduced number of parameters while preserving key features.
   - Compression results can vary based on the architecture, training data, and compression ratios.
   - Autoencoders have the potential to achieve high compression while maintaining image quality, making them valuable in applications like image storage, transmission, and reconstruction.

In summary, as a student, I've explored autoencoder architectures for image compression, understanding their role in reducing image data while preserving essential information. Autoencoders offer versatile solutions for various compression scenarios, and their effectiveness depends on factors like architecture and dataset.