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.

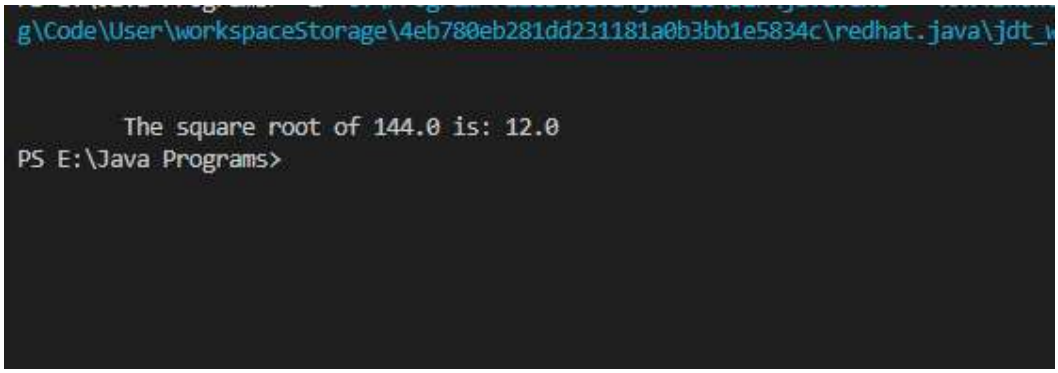
Programmers can define their own packages to bundle group of classes/interfaces, etc. While creating a package, the user should choose a name for the package and include a package statement along with that name at the top of every source file that contains the classes, interfaces, enumerations, and annotation 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:

```
public class Test
{

public static void main(String args[])

{

double number = 144.0;

double squareRoot = Math.sqrt(number);

System.out.println("\n\n\tThe square root of " + number + " is: " + squareRoot);

}

}
```

Output:
The square root of 144.0 is: 12.0

Screenshot:



Conclusion:
Comment on the autoencoder architecture and the Image compression results.
Autoencoders compress images by encoding them into a lower-dimensional latent space and then reconstructing them. The architecture typically involves an encoder to reduce dimensionality and a decoder to reconstruct the image. The size of the latent space directly impacts compression ratio and reconstruction quality. Larger latent spaces preserve more detail but offer lower compression. Effective autoencoders balance this trade-off, achieving good compression while maintaining image fidelity. Performance is often evaluated using metrics like PSNR and SSIM to measure the quality of reconstruction