



Experiment No.5
Aim: Basic programming constructs like branching and looping
Name: Aman Mehtar
Roll no: 32
Date of Performance: 22/8/24
Date of Submission: 29/8/24



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:

In **Greeting.java** file⇒

```
package myPackage;
```

```
public class Greeting {  
    public void sayHello() {  
        System.out.println("Hello from myPackage!");  
    }  
}
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

In **Main.java** file⇒

```
import myPackage.Greeting;  
  
public class Main {  
    public static void main(String[] args) {  
        Greeting greet = new Greeting();  
        greet.sayHello();  
    }  
}
```

Output:

A screenshot of a terminal window with a black background and white text. The text "Hello from myPackage!" is displayed in a monospaced font.

Conclusion:

The autoencoder architecture is a powerful tool for unsupervised learning and image compression. Its ability to learn compact representations of high-dimensional data makes it particularly useful in various applications, including image compression, noise reduction, and feature extraction.

In Java, leveraging libraries like DeepLearning4J allows developers to implement autoencoders effectively, facilitating the development of models that can compress images while maintaining quality. The results can lead to significant reductions in data storage requirements and improved performance in image processing tasks.