



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No. 7
Implement a program using super and final keyword.
Date of Performance:
Date of Submission:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Aim: To implement the concept of super and final keyword.

Objective: To understand the usage of superclass and final method, variables and class

Theory:

super and **final** keywords are two popular and useful keywords in Java. They also play a significant role in dealing with Java programs and their classes. In this chapter, you will learn about how to use super and final within a Java program.

Syntax: `super.<method-name>();`

- Super variables refer to the variable of a variable of the parent class.
- Super() invokes the constructor of immediate parent class.
- Super refers to the method of the parent class

Instance refers an instance variable of the current class by default, but when you have to refer parent class instance variable, you have to use super keyword to distinguish between parent class (here employee) instance variable and current class (here, clerk) instance variable.

What is final in Java?

Final is a keyword in Java that is used to restrict the user and can be used in many respects. Final can be used with:

- Class
- Methods
- Variables

A method declared as final cannot be overridden; this means even when a child class can call the final method of parent class without any issues, but the overriding will not be possible.

Once a variable is assigned with the keyword final, it always contains the same exact value. Again things may happen like this; if a final variable holds a reference to an object then the state of the



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

object can be altered if programmers perform certain operations on those objects, but the variable will always refer to the same object. A final variable that is not initialized at the time of declaration is known as a blank final variable. If you are declaring a final variable in a constructor, then you must initialize the blank final variable within the constructor of the class. Otherwise, the program might show a compilation error.

Code:

```
import java.io.*;
import java.util.*;
class Animal {
    final String type = "Mammal";

    Animal() {
        System.out.println("Animal constructor called.");
    }

    final void sound() {
        System.out.println("Animals make sound.");
    }
}

class Dog extends Animal {
    String breed;

    Dog(String breed) {
        super();
        this.breed = breed;
    }

    void displayInfo() {
        System.out.println("Animal type: " + super.type);
        System.out.println("Dog breed: " + this.breed);
    }
}

public class Main {
    public static void main(String[] args) {
        Dog dog = new Dog("Golden Retriever");
        dog.displayInfo();
        dog.sound();
    }
}
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

}

Output:

```
Animal constructor called.  
Animal type: Mammal  
Dog breed: Golden Retriever  
Animals make sound.  
  
=== Code Execution Successful ===|
```

Conclusion:

The **super** and **final** keywords are essential components of Java's object-oriented programming capabilities. The **super** keyword facilitates interaction between classes in an inheritance hierarchy, allowing subclasses to leverage functionality defined in their superclasses. This promotes code reuse and organization.

On the other hand, the **final** keyword provides mechanisms for enforcing immutability and preventing unintended modifications to variables, methods, and classes. By using these keywords appropriately, developers can create robust, maintainable, and well-structured code, enhancing the overall integrity of their applications.