Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

SAIRAM DIGITAL RESOURCES

YEAR II SEM III

CS8392

OBJECT ORIENTED PROGRAMMING
(Common to CSE, EEE, EIE, ICE, IT)

# UNIT NO 2

## INHERITANCE AND INTERFACES

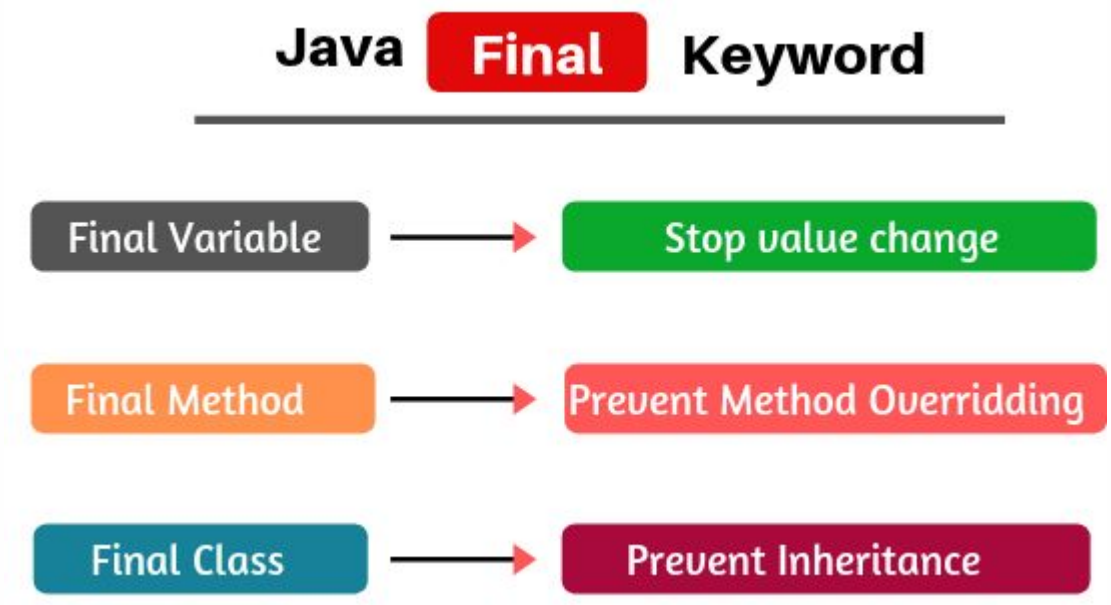
### 2.4 FINAL METHODS AND CLASSES

COMPUTER SCIENCE & ENGINEERING

# final keyword

- final is a non-access modifier for Java elements
- final keyword can be used along with variables, methods and classes.
- Final keyword can be used in different contexts,
  - final variable
  - final method
  - final class

# final keyword



# final variables

- Any variable either a member variable or local variable (declared inside method or block) **prefixed with final keyword** is called final variable.

Final Variable → To create constant variables

- final variables act as constants. A final variable is different from a constant, as the value of the final variable is not necessarily known at the compile time.
- The value of a final variable cannot be changed once it is initialized.
  i.e When a variable is declared with final keyword, its value can't be modified.

# final variables (Contd..)

## Initializing a final variable

- A final variable must be initialized, otherwise compiler will throw compile-time error.
- A final variable can only be initialized once, either via an initializer or an assignment statement.

**There are three ways to initialize a final variable :**

1. Initialize a final variable when it is declared. If a final variable is not initialized at the time of declaration, it must be initialized inside the constructor of the class in which it is declared. Such a variable is called as blank final variable.
2. A blank final variable can be initialized inside instance-initializer block or inside constructor. If a class has more than one constructor in your class then it must be initialized in all of them,
3. A blank final static variable can be initialized inside static block.

# final variables (Contd..)

```
class Point {
        public final int x=1;        // final variable
        public final int y;          // blank final variable
        public static final int z;; // blank final static variable

        //Constructor
        public Point() {
                y=2;
        }

        //Static Initializer block
        static {
                z=3;
        }

        public static void main(String args[]) {
                Point pt= new Point();
                System.out.println(“ (“ + pt.x + “ ,” + pt.y + “ ,” + z + “ )”);
        }
```

Output:

(1,2,3)

# final methods

- A method with the final keyword is called a final method
- A final method cannot be overridden which means even though a subclass can call the final method of parent class without any issues but it cannot override it.


```
class FinalDemo {
        public final void display() {
                System.out.println(“ This is a final method “);
        }
}
class MainDemo extends FinalDemo {
        public void final display() {
                System.out.println(“ This is a final method overridden”);
        }
        public static void main(String args[]) {
                MainDemo obj= new MainDemo();
                obj.final();
        }
}
```

**Output:**

**display() in MainDemo cannot override display() in FinalDemo**

**public final void display()**

**overridden method is final**

# final class

- When a class is declared with final keyword, it is called a final class
- If a class is marked as final then no class can inherit any feature from the final class.i.e a final class cannot be inherited


```
final class XYZ{
}
class ABC extends XYZ{
   void demo(){
        System.out.println("My Method");
   }
   public static void main(String args[]){
        ABC obj= new ABC();
        obj.demo();
   }
}
```

**Output:**

The type ABC cannot subclass the final class XYZ

# Video Link

https://www.youtube.com/watch?v=louYAgvTsLY