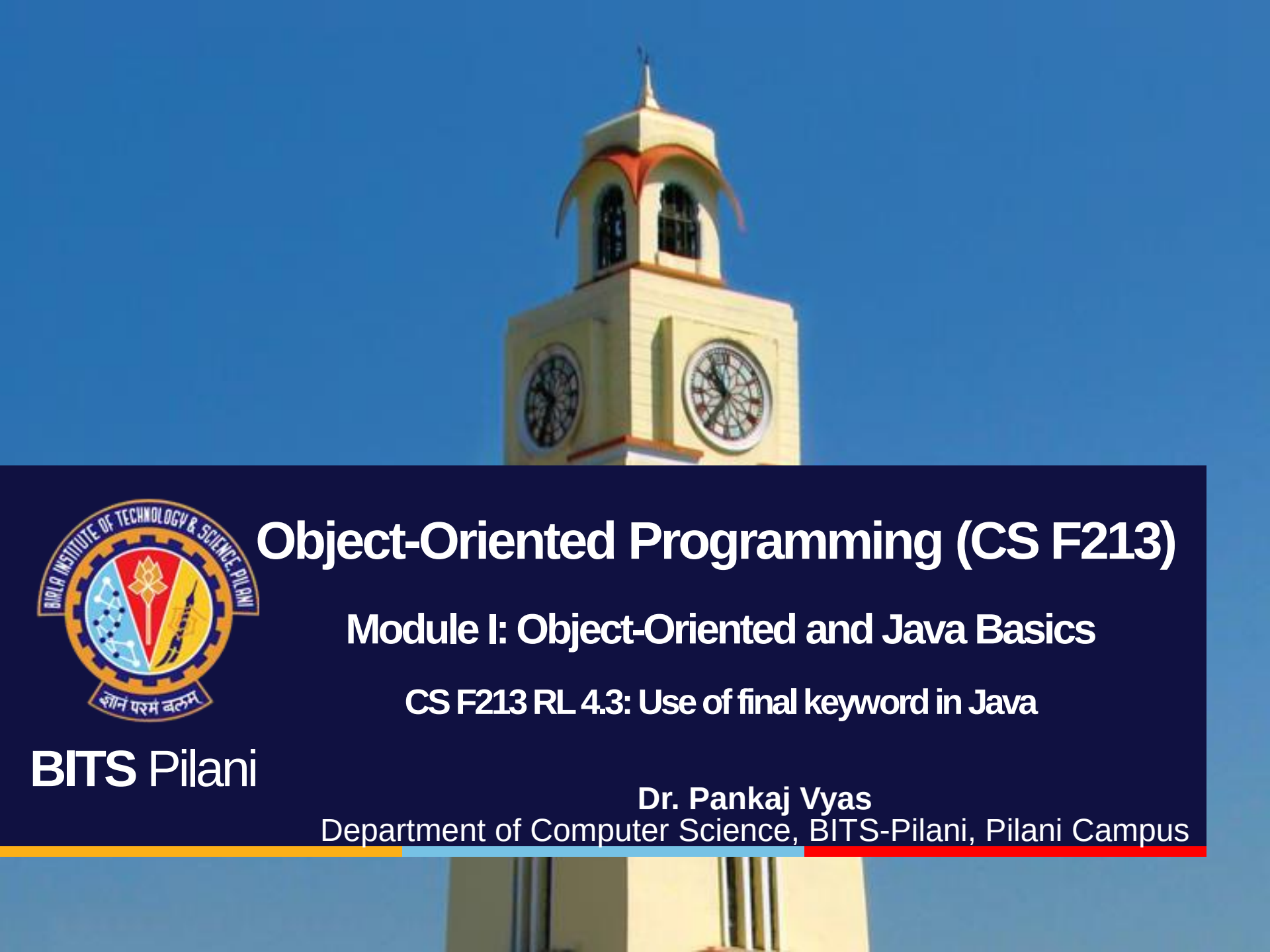


Object-Oriented Programming (CS F213)

Module I: Object-Oriented and Java Basics

CS F213 RL 4.3: Use of final keyword in Java

BITS Pilani

Dr. Pankaj Vyas
Department of Computer Science, BITS-Pilani, Pilani Campus

CS F213 RL 4.3 : Topics

- Use of final keyword in Java
  - final instance fields
  - final methods of class
  - final classes

final instance fields

- ‘final’ instance fields means value of the field is fixed and can not change
- ‘final’ fields have to be explicitly initialized
- Syntax

  <scope> [<static>] <final> <type> variable-name = value

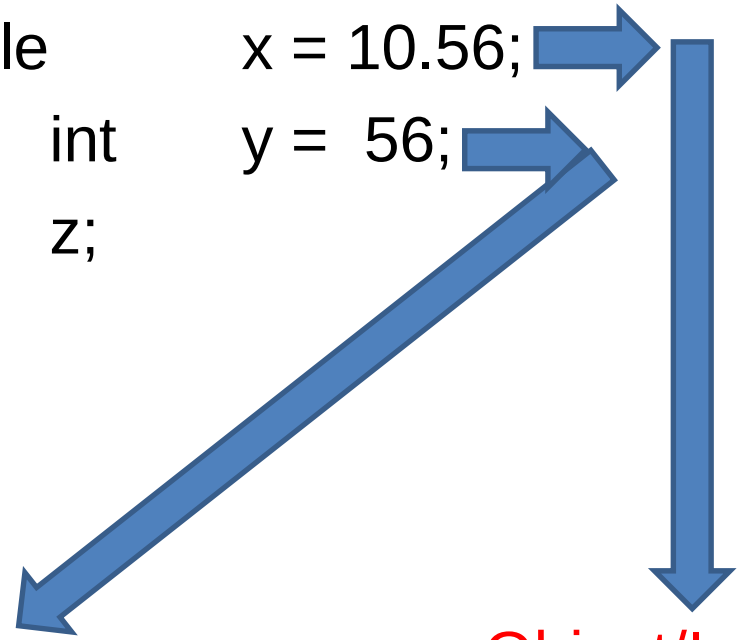
- Example

1.	public	static	final	int	x = 10; → **OK**
2.	private	final	static	double	y = 4.56; → **OK**
3.	public	final	static	double	z; → **Not OK**
4.	private	static	final	float	d; → **Not OK**

Results in Compile-Time Errors

‘final’ instance Fields Example

```
class XYZ
{
    private     final   double          x = 10.56;
    public      static  final   int     y =  56;
    private     double          z;
    ……….
    ……….
}// End of XYZ
```



class/static field is final

Object/Instance field is final

primitive type variables as final

```
// File Name : Demo.java
class Circle
{
        private     double      radius;
        // Constructor Method
        Circle(double radius)
        {
                this.radius = radius;
        }// End of Constructor Methods
        // Method to Get Radius
        public      double      getRadius()
        {
                return this.radius;
        }
        // Method to Set Radius
        public      void        setRadius(double radius)
        {
                this.radius = radius;
        }
        // Method to Get Area of Circle
        public      double      area()
        {
                return 3.1456 * radius * radius;
        }
        // Method to Get Perimeter of Circle
        public      double      perimeter()
        {
                return 2 * 3.1456 * radius;
        }
}// End of class Demo
```

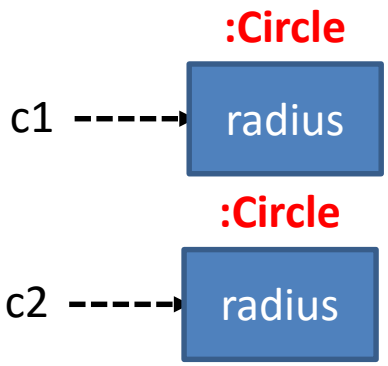
```
//Driver Class
class Test
{
        public      static void main(String s[])
        {
                final       int x = 40;
                x = 20;
        }// End of Methods
}// End of Test class
```

```
F:\>javac Demo.java
Demo.java:37: cannot assign a value to
final variable x
                x = 20;
                ^
1 error
```

class type variables as final

```
// File Name : Demo.java
class Circle
{
        private     double      radius;
        // Constructor Method
        Circle(double radius)
        {
                this.radius = radius;
        }// End of Constructor Methods
        // Method to Get Radius
        public      double      getRadius()
        {
                return this.radius;
        }
        // Method to Set Radius
        public      void        setRadius(double radius)
        {
                this.radius = radius;
        }
        // Method to Get Area of Circle
        public      double      area()
        {
                return 3.1456 * radius * radius;
        }
        // Method to Get Perimeter of Circle
        public      double      perimeter()
        {
                return 2 * 3.1456 * radius;
        }
}// End of class Demo
```

```
//Driver Class
class Test
{
        public      static void main(String s[])
        {
            Circle c1 = new Circle(10);
            Circle c2 = new Circle(20);
        }// End of Methods
}// End of Test class
```



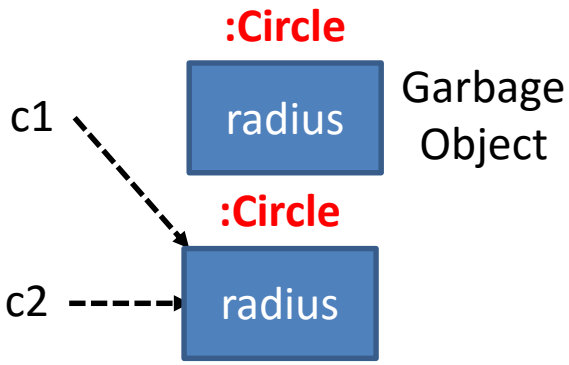
class type variables as final

```
// File Name : Demo.java
class Circle
{
        private     double      radius;
        // Constructor Method
        Circle(double radius)
        {
                this.radius = radius;
        }// End of Constructor Methods
        // Method to Get Radius
        public      double      getRadius()
        {
                return this.radius;
        }
        // Method to Set Radius
        public      void        setRadius(double radius)
        {
                this.radius = radius;
        }
        // Method to Get Area of Circle
        public      double      area()
        {
                return 3.1456 * radius * radius;
        }
        // Method to Get Perimeter of Circle
        public      double      perimeter()
        {
                return 2 * 3.1456 * radius;
        }
}// End of class Demo
```

```
//Driver Class
class Test
{
        public      static void main(String s[])
        {
            Circle c1 = new Circle(10);
            Circle c2 = new Circle(20);

            c1      =       c2;

        }// End of Methods
}// End of Test class
```

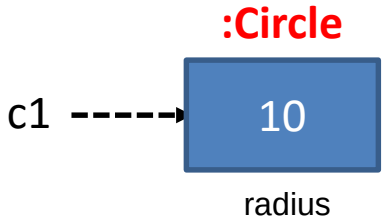


class type variables as final

```
// File Name : Demo.java
class Circle
{
        private     double      radius;
        // Constructor Method
        Circle(double radius)
        {
                this.radius = radius;
        }// End of Constructor Methods
        // Method to Get Radius
        public      double      getRadius()
        {
                return this.radius;
        }
        // Method to Set Radius
        public      void        setRadius(double radius)
        {
                this.radius = radius;
        }
        // Method to Get Area of Circle
        public      double      area()
        {
                return 3.1456 * radius * radius;
        }
        // Method to Get Perimeter of Circle
        public      double      perimeter()
        {
                return 2 * 3.1456 * radius;
        }
}// End of class Demo
```

```
//Driver Class
class Test
{
        public      static void main(String s[])
        {
            final Circle c1 = new Circle(10);

        }// End of Methods
}// End of Test class
```

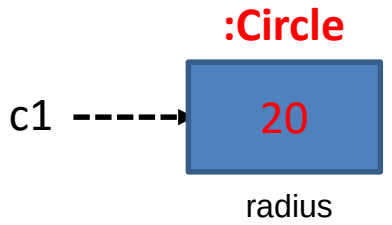


class type variables as final

```
// File Name : Demo.java
class Circle
{
        private     double      radius;
        // Constructor Method
        Circle(double radius)
        {
                this.radius = radius;
        }// End of Constructor Methods
        // Method to Get Radius
        public      double      getRadius()
        {
                return this.radius;
        }
        // Method to Set Radius
        public      void        setRadius(double radius)
        {
                this.radius = radius;
        }
        // Method to Get Area of Circle
        public      double      area()
        {
                return 3.1456 * radius * radius;
        }
        // Method to Get Perimeter of Circle
        public      double      perimeter()
        {
                return 2 * 3.1456 * radius;
        }
}// End of class Demo
```

```
//Driver Class
class Test
{
        public      static void main(String s[])
        {
            final Circle c1 = new Circle(10);
            c1.setRadius(20);

        }// End of Methods
}// End of Test class
```



OK

class type variables as final

```
// File Name : Demo.java
class Circle
{
        private     double      radius;
        // Constructor Method
        Circle(double radius)
        {
                this.radius = radius;
        }// End of Constructor Methods
        // Method to Get Radius
        public      double      getRadius()
        {
                return this.radius;
        }
        // Method to Set Radius
        public      void        setRadius(double radius)
        {
                this.radius = radius;
        }
        // Method to Get Area of Circle
        public      double      area()
        {
                return 3.1456 * radius * radius;
        }
        // Method to Get Perimeter of Circle
        public      double      perimeter()
        {
                return 2 * 3.1456 * radius;
        }
}// End of class Demo
```

```
//Driver Class
class Test
{
        public      static void main(String s[])
        {
            final Circle c1 = new Circle(10);
            c1.setRadius(20);

            Circle c2 = new Circle(5);
            c1 = c2;   // Compile-Time Error

        }// End of Methods
}// End of Test class
```

```
F:\>javac Demo.java
Demo.java:39: cannot assign a value to
final variable c1
                c1 = c2;
                ^
1 error
```

Method parameters / arguments as final

- You can declare method parameters as 'final' also

```
// File Name: Demo.java
class Test
{
        public static void sum(final int a, int b)
        {
                a = 56;
                System.out.println(a+b);
        }
        public static void main(String args[])
        {
                sum(10,5);
        }
}
```

final Method Argument

Erroneous Statement

```
F:\>javac Demo.java
Demo.java:6: final parameter a may not be assigned
                a = 56;
                ^
1 error
```

final classes

- final class means class definition is final and can not have sub-classes

```
final class X
{
}// End of class X

class Y extends X
{
}// End of class Y
```

```
F:\>javac Demo.java
Demo.java:4: cannot inherit
from final X
class Y extends X
                ^
1 error
```

final Methods

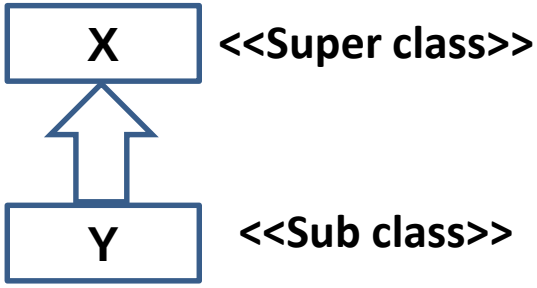
- 'final' methods means the implementation of the method is final. Sub classes can not override the method.
- 'final' and 'abstract' keywords can not be used together for a method

```
class X
{
        public final abstract void doS() {}
} // End of class X
```

<<illegal combination of modifiers: abstract and final>>

final Methods : Example

```
// File Name: Demo.java
class X
{
        public void doS() { }
} // End of class X
class Y extends X
{
        public void doS() { }
} // End of class Y
```



class Y overrides the doS() method of class X

```
// File Name: Demo.java
class X
{
        public final void doS()
{ }
} // End of class X
class Y extends X
{
        public void doS() { }
} // End of class Y
```

```
F:\>javac Demo.java
Demo.java:8: doS() in Y cannot
override doS() in X; overridden
method is final
        public void doS() { }
                    ^
1 error
```

Thank You