VIT-AP UNIVERSITY

Course code : **CSE2005**

Course title : **Object Oriented Programming**

Final keyword

Objectives

This session will give the knowledge about

- final Keyword

final Keyword

- The **final** keyword used in context of behavioral restriction on:
  - variables
  - methods
  - classes
- Using final on variables to make them behave as constants which we have seen in earlier module.

final Keyword

- When a variable is made final – it can be initialized only once either by
  - Declaration and initialization
    - **final int x=10;**
  - Using constructor
- System allows you to set the value only once; after which it can't be changed.

final Keyword

```
package vit.demo;

class Sample {
   final double pi;
   public Sample() {
          pi = 3.14;
   }
   public Sample(double pi) {
          this.pi = pi;
   }
}
```

```
public class Main {
      public static void main(String ar[]){
          Sample s1=new Sample();
          Sample s2=new Sample(22/7);
          System.out.println(s1.pi+" "+s2.pi);
      }
}
```

Quiz

```
package vit.demo;

class Sample {
   final double pi=2;
   public Sample() {
          pi = 3;
   }
   public Sample(double pi) {
          this.pi = pi;
   }
}
```

```
public class Main {
      public static void main(String ar[]){
          Sample s1=new Sample();
          Sample s2=new Sample(4);
          s1.pi = 5;
          System.out.println(s1.pi+" "+s2.pi);
      }
}
```

The Role of the Keyword final in Inheritance

The final keyword has two important uses in the context of a class hierarchy. These uses are highlighted as follows:

Using final to Prevent Overriding

- When you would want the subclasses to use the methods as they are defined in the superclass, you can prevent overriding.
- This can be achieved by declaring such critical methods as final.

final Keyword to prevent Overrideing

```
package vit.demo;

class Sample {
   final void display(){
          System.out.println("sample");
   }
}
```

```
public class Main extends Sample {
      void display(){
          System.out.println("main");
      }
      public static void main(String ar[]){
          Sample s1=new Sample();
          s1.display();
      }
}
```

The Role of the Keyword final in Inheritance

Using final to Prevent Inheritance

- Sometimes you will want to prevent a class from being inherited. This can be achieved by preceding the class declaration with final.
- Declaring a class as final implicitly declares all of its methods as final too.
- It is illegal to declare a class as both abstract and final since an abstract class is incomplete by itself and relies upon its subclasses to provide concrete and complete implementations.

final Keyword to prevent Inheritance

```
package vit.demo;

final class Sample {
   void display(){
         System.out.println("sample");
   }
}
```

```
public class Main extends Sample {
      void display(){
         System.out.println("main");
      }
      public static void main(String ar[]){
         Sample s1=new Sample();
         s1.display();
      }
}
```

Quiz

```
package vit.demo;

abstract class Sample {
   final void display(){
          System.out.println("sample");
   }
}
```

```
public class Main extends Sample {
      public static void main(String ar[]){
          Main s1=new Main();
          s1.display();
      }
}
```

Quiz

```
package vit.demo;
 abstract class A{
   abstract final void display();
}
abstract class B extends A{
   abstract void show();
}
```

```
public class Main extends B {
   public static void main(String a[]) {
            System.out.println("main");
   }
}
```

Quiz

```
package vit.demo;
class A{
   final void display();
}
final class B extends A{
   void display() {
   System.out.println("I am main");
   }
}
```

```
public class Main extends B {
   public static void main(String a[]) {
            System.out.println(“main");
   }
}
```

Summary

We have discussed about

- final Keyword