BITS Pilani

Object-Oriented Programming (CS F213)

Module III: Inheritance and Polymorphism in Java

CS F213 RL 10.5: Local Classes

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

CS F213 RL 10.5 : Topics

- Local Classes

Local Classes

- Class With-in The Boundary of a Method of some other Class
- Two Forms : 1. Local Class
  2. Anonymous Inner Classes

```
class A
{
        public   void     show()
        {
                class X
                {
                } // End of class X
                class Y extends X
                {
                } // End of class Y
        }// End of Method
}// End of class A
```

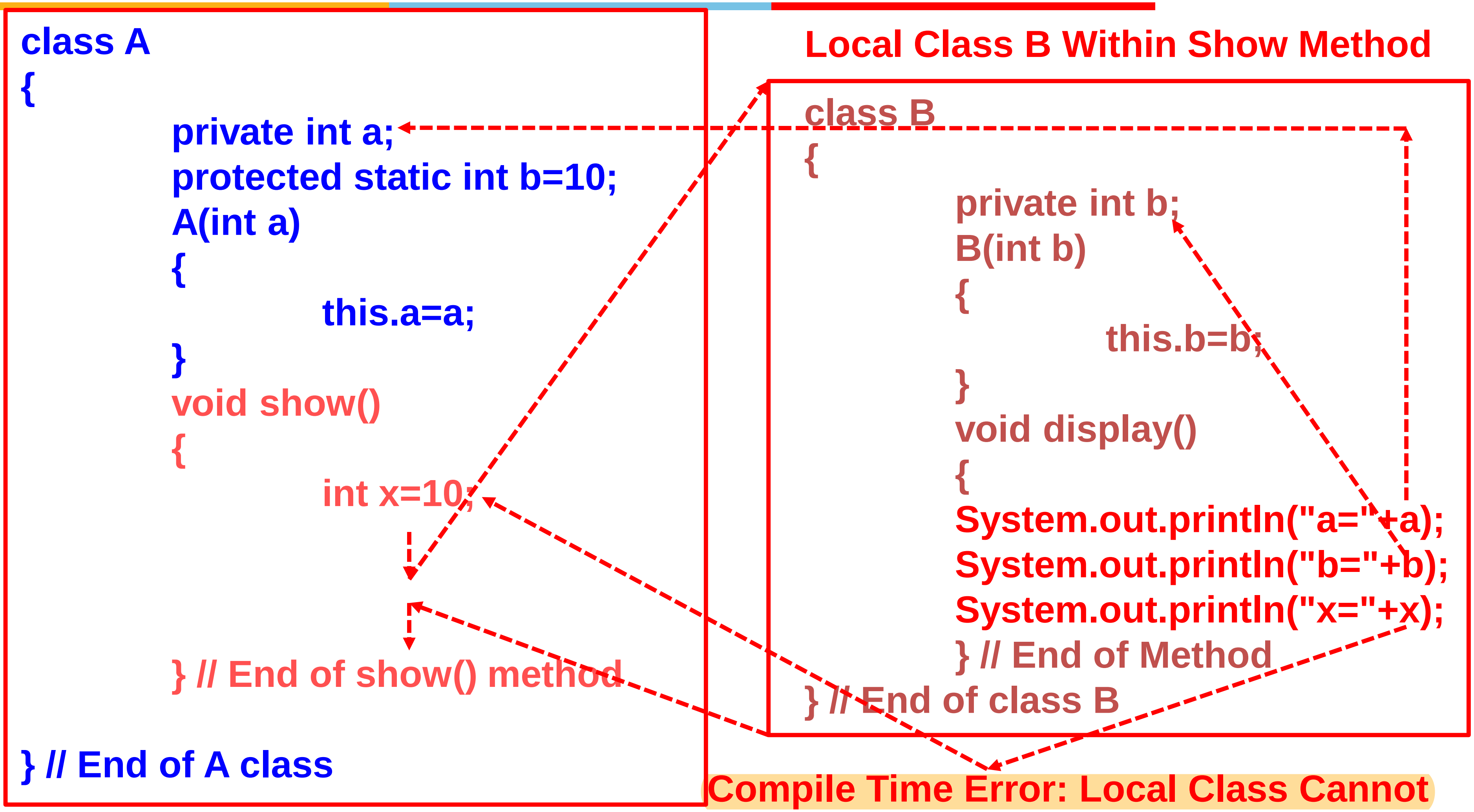
Local Class X

Local Class Y

Local Classes

- Local Classes are Visible Only in the Methods in which they are defined. [No Scope Access Modifier such as public, protected can be used for Local Classes]
- Local Classes can Only Use ‘final’ variables from its enclosing method.
- However, Local Classes can Access instance-fields and object methods of the enclosing class even if they are private.

Local Classes .. Example



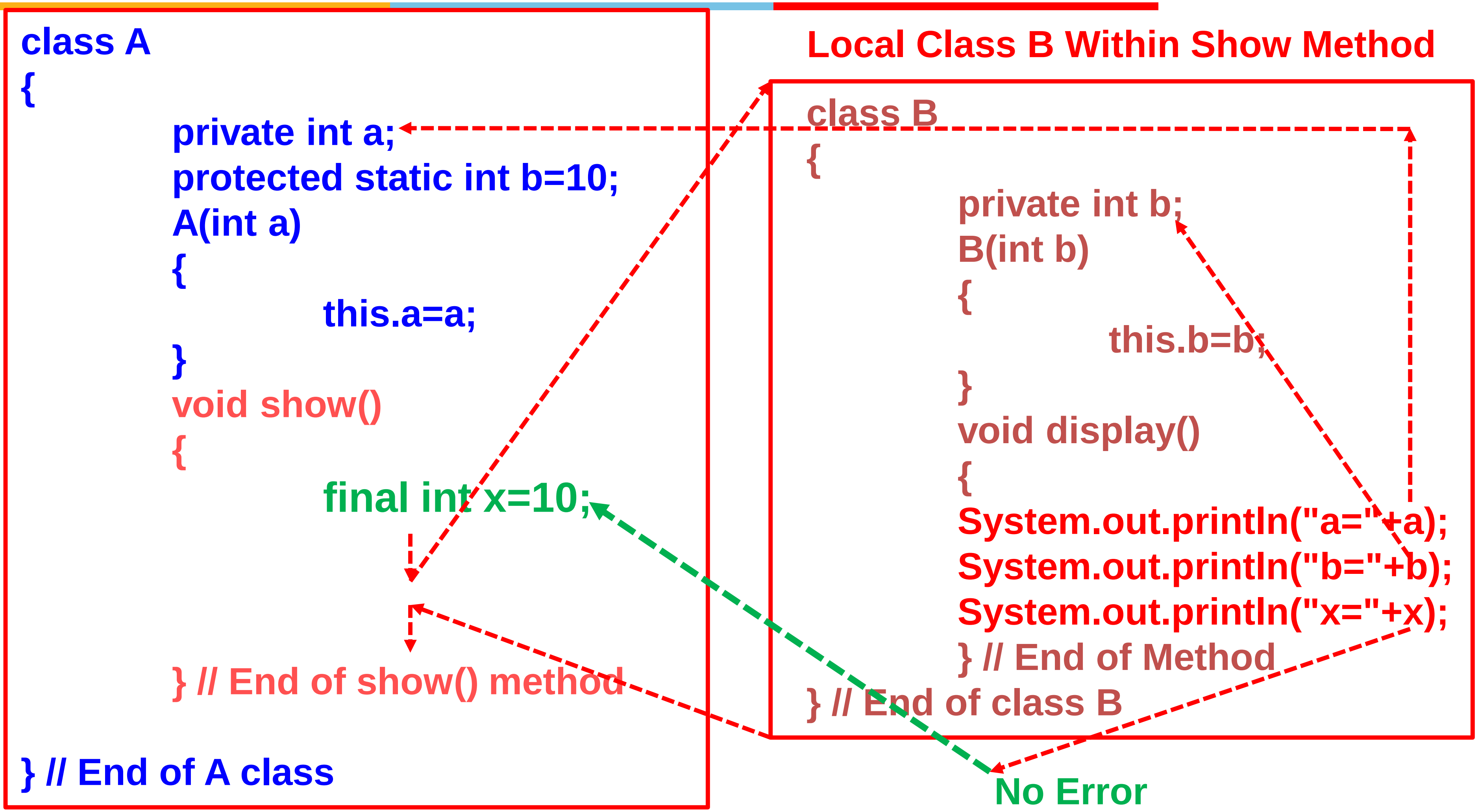
Local Class B Within Show Method

```
class A
{
        private int a;
        protected static int b=10;
        A(int a)
        {
                this.a=a;
        }
        void show()
        {
                int x=10;



        } // End of show() method

} // End of A class
```

```
class B
{
        private int b;
        B(int b)
        {
                this.b=b;
        }
        void display()
        {
        System.out.println("a="+a);
        System.out.println("b="+b);
        System.out.println("x="+x);
        } // End of Method
} // End of class B
```

Compile Time Error: Local Class Cannot Access non-final Variables From the Enclosing Method

Local Classes .. Example 1



```
class A
{
        private int a;
        protected static int b=10;
        A(int a)
        {
                this.a=a;
        }
        void show()
        {
                final int x=10;



        } // End of show() method

} // End of A class
```

Local Class B Within Show Method

```
class B
{
        private int b;
        B(int b)
        {
                this.b=b;
        }
        void display()
        {
        System.out.println("a="+a);
        System.out.println("b="+b);
        System.out.println("x="+x);
        } // End of Method
} // End of class B
```

No Error

Local Class Can Access Only final Variables From the Enclosing Method

Local Classes .. Example 2

```
// File Name : inner.java
class Test
{
    public  static      void        main(String args[])
    {
            final       int  a1 = 10;



            new A(20).show();
            print();
     }// End of Method
     static void print()
     {
            /*
            A a1 = new A(30);
            a1.show();
            */
     } // End of Method
}// End of class Test
```

```
class A
{
            private int a;
            private int b;
            int c;
            A(int a)
            {
                        this.a =a;
                        b = a+20;
                        c = a+40;
            }
            void show()
            {
            System.out.println("a1="+a1);
            System.out.println("a="+a);
            System.out.println("b="+b);
            System.out.println("c="+c);
            } // End of Method
} //End of class A
```

Anonymous Inner Classes

- Local Classes Without a Name (Another Form of Local Classes)
- Can either extend an Existing Concrete or Abstract class or Can implement an Existing Interface
- Only one instance of an Anonymous Inner class can be created
- Whole body of an Anonymous Inner class is defined in a single statement ending with semi-colon (;)
- Frequently used for Writing a GUI and an Event Handling type of Applications

Anonymous Inner Classes : Syntax

- **If extending a class (Either Concrete or Abstract)**

```
super-class-name  reference-variable = new super-class-name()
{
            …………..
                                    Body of the Class
            …………..
};
```

- **If implementing an interface**

```
interface-name  reference-variable = new interface-name()
{
            …………..
                                    Body of the Class
            …………..
};
```

Anonymous Class : Example 1 (Inner Class Extending a Class)

```
// File Name : inner.java
class A
{
        private int a;
        A(int a)
        {
                this.a =a;
        }
        void show()
        {
                System.out.println("a="+a);
        } // End of show()
}// End of class A
```

Anonymous Class : Example 1…

```
class Test
{
        public static void main(String args[])
        {
                A a1 = new A(20)
                {
                        public void show()
                        {
                            super.show();
                            System.out.println("Hello");
                        }
                        public void display()
                        {
                            System.out.println("Hi");
                        }
                };
                a1.show();
                //a1.display();
        }// End of Method
}// End of Class Test
```

No Semicolon

Body of Anonymous Sub-class of A

Calling show () Method of Inner Class

Anonymous Class : Example 2 (Inner Class Implementing an Interface)

```
// File Name : Inner2.java
interface X
{
            int sum(int a,int b);
            int mul(int x,int y);
} // End of Interface X
class Test
{
            public      static      void        main(String args[])
            {
                        X           x1          =           new         X()
                        {
                                    public      int         show(int a, int b)
                                    {
                                                return a+b;
                                    }// End of Method
                                    public      int         mul(int a, int b)
                                    {
                                                return a*b;
                                    }// End of Method
                        }; // End of class
```

No Semicolon

Body of an Anonymous class implementing an interface X

Anonymous Class : Example 2 … (Inner Class Implementing an Interface)

```
                    System.out.println(x1.show(10,30));

                    System.out.println(x1.mul(10,30));
          }// End of main() Method
}// End of class Test
```

Thank You