class Demo{

final int MAX\_VALUE=99;

void myMethod(){

MAX\_VALUE=101;

}

public static void main(String args[]){

Demo obj=new Demo();

obj.myMethod();

}

}

**Output:**

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

The final field Demo.MAX\_VALUE cannot be assigned

at beginnersbook.com.Demo.myMethod(Details.java:6)

at beginnersbook.com.Demo.main(Details.java:10)

**FINAL METHOD**

class XYZ{

final void demo(){

System.out.println("XYZ Class Method");

}

}

class ABC extends XYZ{

void demo(){

System.out.println("ABC Class Method");

}

public static void main(String args[]){

ABC obj= new ABC();

obj.demo();

}

}

The above program would throw a compilation error, however we can use the parent class final method in sub class without any issues. Lets have a look at this code: This program would run fine as we are not[**overriding**](http://beginnersbook.com/2014/01/method-overriding-in-java-with-example/) the final method. That shows that final methods are[**inherited**](http://beginnersbook.com/2013/05/java-inheritance-types/) but they are not eligible for overriding.

class XYZ{

final void demo(){

System.out.println("XYZ Class Method");

}

}

class ABC extends XYZ{

public static void main(String args[]){

ABC obj= new ABC();

obj.demo();

}

}

**Output:**

XYZ Class Method

**3) final class**

We cannot extend a final class. Consider the below example:

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

**ENHANCED FOR LOOP**

import java.util.Scanner;

class SumDemo{

public static void main(String args[]){

Scanner scanner = new Scanner(System.in);

int[] array = new int[10];

int sum = 0;

System.out.println("Enter the elements:");

for (int i=0; i<10; i++)

{

array[i] = scanner.nextInt();

}

for( int num : array) {

sum = sum+num;

}

System.out.println("Sum of array elements is:"+sum);

}

}

FOR LOOP

|  |
| --- |
| public class SumArrayWithForLoop { |

|  |  |
| --- | --- |
| 04 |  |

|  |  |
| --- | --- |
| 05 | public static void main(String[] args) { |

|  |  |
| --- | --- |
| 06 |  |

|  |  |
| --- | --- |
| 07 | // array to sum |

|  |  |
| --- | --- |
| 08 | int[] numbers = new int[]{ 10, 10, 10, 10}; |

|  |  |
| --- | --- |
| 09 |  |

|  |  |
| --- | --- |
| 10 | int sum = 0; |

|  |  |
| --- | --- |
| 11 |  |

|  |  |
| --- | --- |
| 12 | for (int i=0; i < numbers.length ; i++) { |

|  |  |
| --- | --- |
| 13 | sum = sum + numbers[i]; |

|  |  |
| --- | --- |
| 14 | } |

|  |  |
| --- | --- |
| 15 |  |

|  |  |
| --- | --- |
| 16 | System.out.println("Sum value of array elements is : " + sum); |

|  |  |
| --- | --- |
| 17 |  |

|  |  |
| --- | --- |
| 18 | } |

|  |  |
| --- | --- |
| 19 |  |

|  |  |
| --- | --- |
| 20 | } |