**Meth od hiding factors (MHF) :-**

1. if a subclass defines a static method with the same signature as a static method in the super class.
2. In such a case, the method in the subclass hides the one in the superclass.
3. The mechanism is known as **method hiding**.
4. It happens because static methods are resolved at compile time.

class A

**{**

    static void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{**

    static void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q01\_Method\_Hiding

**{**

    public static void main**(**String args**[])**

**{**

        // 01

        A a **=** **new** A**();**

        a**.**show**();**

        // 02

        B b **=** **new** B**();**

        b**.**show**();**

        // 03

        A a1 **=** **new** B**();**

        a1**.**show**();**

**}**

**}**

// 01

Class A

// 02

Class B

// 03

Class A

class A

**{**

    static void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{}**

class Q02\_Method\_Hiding

**{**

    public static void main**(**String args**[])**

**{**

        // 01

        A a **=** **new** A**();**

        a**.**show**();**

        // 02

        B b **=** **new** B**();**

        b**.**show**();**

        // 03

        A a1 **=** **new** B**();**

        a1**.**show**();**

**}**

**}**

// 01

Class A

// 02

Class A

// 03

Class A

class A

**{}**

class B **extends** A

**{**

    static void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q03\_Method\_Hiding

**{**

    public static void main**(**String args**[])**

**{**

        // 01

        // A a = new A();

        // a.show();

        // 02

        // B b = new B();

        // b.show();

        // 03

        A a1 **=** **new** B**();**

        a1**.**show**();**

**}**

**}**

// 01

 error: cannot find symbol

                a.show();

                 ^

// 02

Class B

// 03

error: cannot find symbol

                a1.show();

                  ^

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q04\_Method\_OverLoading

**{**

    public static void main**(**String args**[])**

**{**

        // 01

        A a **=** **new** A**();**

        a**.**show**();**

        // 02

        B b **=** **new** B**();**

        b**.**show**();**

        // 03

        A a1 **=** **new** B**();**

        a1**.**show**();**

**}**

**}**

// 01

Class A

// 02

Class B

// 03

Class B

class A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class A"**);**

**}**

**}**

class B **extends** A

**{}**

class Q05\_Method\_OverLoading

**{**

    public static void main**(**String args**[])**

**{**

        // 01

        A a **=** **new** A**();**

        a**.**show**();**

        // 02

        B b **=** **new** B**();**

        b**.**show**();**

        // 03

        A a1 **=** **new** B**();**

        a1**.**show**();**

**}**

**}**

// 01

Class A

// 02

Class A

// 03

Class A

Reference Variable Compile Time par check hota he.

Object Rum Time par check hota he.

class A

**{}**

class B **extends** A

**{**

    void show**()**

**{**

        System**.**out**.**println**(**"Class B"**);**

**}**

**}**

class Q06\_Method\_OverLoading

**{**

    public static void main**(**String args**[])**

**{**

        // 01

        // A a = new A();

        // a.show();

        // 02

        B b **=** **new** B**();**

        b**.**show**();**

        // 03

        // A a1 = new B();

        // a1.show();

**}**

**}**

// 01

error: cannot find symbol

    a.show();

     ^

// 02

Class B

// 03

error: cannot find symbol

    a1.show();

^