**public class Baseclass**

**{**

**int x = 20;**

**// Overridden method.**

**void msg()**

**{**

**System.out.println("Base class method");**

**}**

**}**

**public class Childclass extends Baseclass**

**{**

**int x = 50;**

**int y = 100;**

**// Overriding method.**

**void msg()**

**{**

**System.out.println("Child class first method");**

**}**

**void msg2()**

**{**

**System.out.println("Child class second method");**

**}**

**}**

**public class MyTest extends Childclass {**

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

**{**

**Childclass obj = new Childclass();**

**System.out.println("Value of x: " +obj.x); // x of class Childclass is called.**

**obj.msg(); // msg() of Childclass is called.**

**obj.msg2(); // msg2() of Childclass is called.**

**Baseclass obj2 = new Childclass();**

**System.out.println("Value of x: " +obj2.x); // x of Baseclass is called.**

**// System.out.println("Value of y: " +obj2.y); // Error because y does not exist in Baseclass.**

**obj2.msg(); // msg() of Childclass is called.**

**// obj2.msg2(); // Error because the method msg2() does not exist in Baseclass.**

**}**

**}**

**//New Example**

**public class Hello**

**{**

**// Declare an instance block.**

**{**

**show();**

**}**

**Hello()**

**{**

**System.out.println("Hello constructor");**

**show();**

**}**

**void show()**

**{ System.out.println("Hello method");**

**}**

**}**

**public class Hi extends Hello**

**{**

**Hi()**

**{**

**System.out.println("Hi constructor");**

**}**

**void show() { // Override the show() method.**

**System.out.println("Hi method");**

**}**

**}**

**public class TestHelloHi extends Hi**

**{**

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

**{**

**Hi obj = new Hi();**

**obj.show(); // show() method of Hi class is called.**

**// Superclass reference is equal to child class object.**

**Hello obj1 = new Hi();**

**obj1.show();**

**}**

}

**// new example**

**package inheritancePractice;**

**public class Animal**

**{**

**void food()**

**{**

**System.out.println("What kind of food do lions eat?");**

**}**

**}**

**public class Lion extends Animal**

**{**

**void food(int x)**

**{**

**System.out.println("Lions eat flesh");**

**}**

**}**

**public class LionTest extends Lion**

**{**

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

**{**

**Animal a = new Lion();**

**a.food(); // food() method of class Animal is called.**

**// a.food(20); // Compile time error.**

**Lion l = new Lion();**

**l.food(); // food() method of class Lion is called.**

**l.food(10); // food() method of class Lion is called.**

**} }**

**//new example**

**package inheritancePractice;**

**public class AA**

**{**

**int x = 20;**

**int y = 30;**

**void msg1()**

**{**

**System.out.println("I am msg1 in class AA");**

**}**

**void msg2()**

**{**

**System.out.println("I am msg2 in class AA");**

**}**

**}**

**package inheritancePractice;**

**public class BB extends AA**

**{**

**int y = 50;**

**int z = 60; // Overridding method.**

**void msg2()**

**{**

**System.out.println("I am msg2 in class BB");**

**}**

**void msg3()**

**{**

**System.out.println("I am msg3 in class BB");**

**}**

**}**

**public class Scenario1**

**{**

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

**{**

**// Scenario 1.**

**// Create an object of class AA.**

**AA a = new AA(); // 'a' is reference variable of class A and pointing to the object of class AA. Therefore, superclass object reference a is eligible to call only A.**

**System.out.println("Value of x: " +a.x); // x of class AA is called.**

**System.out.println("Value of y: " +a.y); // y of class AA is called.**

**// System.out.println("Value of z: " +a.z); // // Error because z does not exist in AA. // Call msg1() and msg2() methods using reference variable a.**

**a.msg1(); // msg1 of class AA is called.**

**a.msg2(); // msg2 of class AA is called.**

**// a.msg3(); //Error because the method msg3 does not exist in AA.**

**}**

**}**

package inheritancePractice;

public class Scenario2

{

public static void main(String[] args)

{

// Create an object of class BB.

BB b = new BB(); // Here, 'b' is reference variable of class BB and pointing to the object of class BB.

System.out.println("Value of x: " +b.x); // x of class BB is called because by default, x of class AA is available in class BB through inheritance.

System.out.println("Value of y: " +b.y); // y of class BB is called, not of class AA because the object is created for class BB.

System.out.println("Value of z: " +b.z); // z of class BB is called.

b.msg1(); // msg1 of class BB is called because it is available in class BB by default.

b.msg2(); // msg2 of class BB is called, not of class AA because an object is created for class BB.

b.msg3();

}

}

public class Scenario3

{

public static void main(String[] args)

{

// Superclass reference is equal to child class object.

AA a = new BB(); // 'a' is reference variable of class AA but pointing to the object of class BB.

System.out.println("Value of x: " +a.x); // x of class AA is called.

System.out.println("Value of y: " +a.y); // y of class AA is called.

// System.out.println("Value of z: " +a.z); // Error because z does not exist in AA.

a.msg1(); // msg1 of class BB is called because it is available by default in class BB.

a.msg2(); // The overridden msg2 of class BB is called because object is created for class BB.

// a.msg3(); // Error because msg3 does not exist in AA. msg3() is newly created method in class BB that cannot be called by using reference variable 'a' of superclass and pointing to the object of subclass.

}

}

public class Scenario4

{

public static void main(String[] args)

{

AA a = new AA();

BB b = new BB();

a = b;

System.out.println("Value of x: " +a.x);

System.out.println("Value of y: " +a.y);

// System.out.println("Value of z: " +a.z); // Error because z does not exist in class AA.

a.msg1();

a.msg2();

// a.msg3(); // Error msg3 of class AA not exist.

}

}

package inheritancePractice;

public class Scenario6

{

public static void main(String[] args)

{

AA a = new BB();

BB b = new BB();

b = (BB)a; // It looks like superclass assigned to subclass but it translates internally to // BB b=new BB(); // It is equivalent to 2nd scenario.

System.out.println("Value of x: " +a.x);

System.out.println("Value of y: " +a.y);

// System.out.println("Value of z: " +a.z); // Error because z does not exist in class AA.

a.msg1();

a.msg2();

// a.msg3(); // Compilation error.

}

}