abstract class A

{

abstract void test1();

void test2()

{

//some statements

}

}

class B

{

abstract void test1();

void test2()

{

//some statements

}

}

OUTPUT:

B.java:1: error: B is not abstract and does not override abstract method test1() in B

class B

^

1 error

abstract class C

{

void test1();

void test2()

{

//some statements

}

}

OUTPUT:

C.java:3: error: missing method body, or declare abstract

void test1();

^

1 error

abstract class D

{

abstract void test1();

abstract void test2()

{

//some statements

}

}

OUTPUT:

D.java:4: error: abstract methods cannot have a body

abstract void test2()

^

1 error

abstract class A

{

abstract void test1();

void test2()

{

System.out.println("from test2");

}

}

class E extends A

{

}

OUTPUT:

E.java:9: error: E is not abstract and does not override abstract method test1() in A

class E extends A

^

1 error

abstract class A

{

abstract void test1();

void test2()

{

System.out.println("from test2");

}

}

abstract class F extends A

{

}

abstract class A

{

abstract void test1();

void test2()

{

System.out.println("from test2");

}

}

class G extends A

{

void test1()

{

//some statement

}

}

abstract class A

{

abstract void test1();

void test2()

{

System.out.println("from test2");

}

}

class G extends A

{

void test1()

{

//some statement

}

}

class H

{

public static void main(String[] agrs)

{

A a1 = new A();

G g1 = new G();

System.out.println("done");

}

}

OUTPUT:

H.java:20: error: A is abstract; cannot be instantiated

A a1 = new A();

^

1 error

abstract class A

{

abstract void test1();

void test2()

{

System.out.println("from test2");

}

}

class G extends A

{

void test1()

{

//some statement

}

}

class I

{

public static void main(String[] agrs)

{

A a1 = null;

G g1 = new G();

System.out.println("done");

}

}

OUTPUT:

done

abstract class A

{

abstract void test1();

void test2()

{

System.out.println("from test2");

}

}

class G extends A

{

void test1()

{

//some statements

}

}

class J

{

A obj;

J(A obj)

{

}

void method1(A obj)

{

}

A method2()

{

return null;

}

public static void main(String[] agrs)

{

A a1 = null;

G g1 = new G();

System.out.println("done");

}

}

OUTPUT:

done

abstract class A

{

abstract void test1();

void test2()

{

System.out.println("from A.test2");

}

}

class B extends A

{

void test1()

{

System.out.println("from B.test1");

}

}

class K

{

public static void main(String[] agrs)

{

B b1 = new B();

b1.test1();

b1.test2();

System.out.println("done");

}

}

OUTPUT:

from B.test1

from A.test2

done

abstract class A

{

abstract void test1();

void test2()

{

System.out.println("from A.test2");

}

abstract void test3();

}

abstract class B extends A

{

void test1()

{

System.out.println("from B.test1");

}

}

class C extends B

{

void test3()

{

System.out.println("from C.test1");

}

}

class L

{

public static void main(String[] agrs)

{

//B b1 = new B();

C c1 = new C();

c1.test1();

c1.test2();

c1.test3();

System.out.println("done");

}

}

OUTPUT:

from B.test1

from A.test2

from C.test1

done

abstract class A

{

A()

{

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

}

abstract void test1();

void test2()

{

System.out.println("from A.test2");

}

abstract void test3();

}

abstract class B extends A

{

B(int i)

{

System.out.println("B(int)");

}

void test1()

{

System.out.println("from B.test1");

}

}

class C extends B

{

C()

{

super(90);

System.out.println("C()");

}

void test3()

{

System.out.println("from C.test3");

}

}

class M

{

public static void main(String[] agrs)

{

//B b1 = new B();

C c1 = new C();

c1.test1();

c1.test2();

c1.test3();

System.out.println("done");

}

}

OUTPUT:

A()

B(int)

C()

from B.test1

from A.test2

from C.test3

done