

```

1) class J
{
    {System.out.println("IIB");
    }
    public static void main(String[] args)
    {System.out.println("main");
    }
    static
    {System.out.println("SIB");
    }
}

2) class K
{
    {System.out.println("IIB");
    }
    public static void main(String[] args)
    {
        K k1=new K();
        System.out.println("main method");
    }
    static
    {System.out.println("SIB");
    }
}

3) class L
{
    void test()
    {System.out.println("test method");
    }
    public static void main(String[] args)
    {
        L l1=new L();
        System.out.println("main method");
        l1.test();
    }
    static
    {System.out.println("SIB");
    }
    {System.out.println("IIB");
    }
}

4) class M
{
    static M test()
    {
        return new M();
    }
    public static void main(String[] args)
    {
        M i=test();
        i.test1();
    }
    void test1()
    {
        System.out.println("Success");
    }
}

5) class S
{
    int i;
    static void test1(int x)
    {
        x=10;
    }
    static void test2( S s1)
    {
        s1.i=20;
    }
    public static void main(String[] args)
    {
        S s1=new S();
        s1.i=30;
        System.out.println("A:"+s1.i);
        S s2=new S();
        test1(s1.i);
        System.out.println("B:"+s1.i);
        test2(s1);
        System.out.println("C:"+s1.i);
    }
}

```

```

}

6) class N
{
    int i;
    static N test()
    {
        N t1=new N();
        return t1;
    }
    public static void main(String[] args)
    {
        N obj=test();
        System.out.println(obj.i);
    }
}

7) class O
{
    O()
    {System.out.println("I am constructor");
    }
    public static void main(String[] args)
    {
        O o1=new O();
        System.out.println("Success");
    }
}

8) class P
{
    P()
    {System.out.println("I am constructor");
    }
    public static void main(String[] args)
    {
        System.out.println("Success");
    }
}

9) class Q
{
    Q()
    {
        System.out.println("Constructor");
    }
    public static void main(String[] args)
    {
        Q obj=new Q();
        System.out.println("Success");
    }
    static
    {
        System.out.println("SIB");
    }
}

10) class R
{
    R(int i)
    {System.out.println("I can overload");
    }
    public static void main(String[] args)
    {
        R r1=new R();
        System.out.println("Hello World!");
    }
    {
        System.out.println("From IIB");
    }
}

11) class S
{
    S(int i)
    {System.out.println("int()");
    }
    S(int i, int j)
    {
        System.out.println("int(),int()");
        System.out.println("success");
    }
    public static void main(String[] args)
    {
        S obj=new S(10);
    }
}

12) class T
{
    T(int i)
    {

```

```

        System.out.println("Summer is
        started");
    }
    T(int i,int j)
    {
        this();
        System.out.println("Yes, started");
    }
    public static void main(String[] args)
    {
        T obj=new T(10,10);
    }
}
13) class U
{
    U()
    {
        System.out.println("N()");
    }
    U(int i)
    {
        this();
        System.out.println("N(int)");
    }
    public static void main(String[] args)
    {
        U obj=new U();
        System.out.println("=====");
        U obj2=new U(10);
        System.out.println("=====");
    }
}
14) class V
{
    V()
    {
        this(10);
        System.out.println("U()");
    }
    V(int i)
    {
        this();
        System.out.println("U(int)");
    }
    public static void main(String[] args)
    {
        V obj=new V();
        System.out.println("Success");
    }
}
15) class S
{
    S()
    {
        System.out.println("T()");
    }
    S(int i)
    {
        S s1=new S();
        System.out.println("T()");
    }
    public static void main(String[] args)
    {
        S obj1=new S();
        System.out.println("-----");
        S obj2=new S(10);
        System.out.println("-----");
    }
}
16) class X
{
    public static void main(String[] args)
    {
        X obj=new X(90);
        System.out.println("Success");
    }
}

```

```

    }
17) class Y
    {
        Y()
        {
            System.out.println("D()");
        }
        {
            System.out.println("IIB1");
        }
        {
            System.out.println("IIB2");
        }
        public static void main(String[] args)
        {
            Y obj=new Y();
            System.out.println("-----");
            Y obj1=new Y();
            System.out.println("-----");
        }
    }
18) class A
    {
        int i;
    }
    class B extends A
    {
        int j;
        public static void main(String[] args)
        {
            B b1=new B();
            System.out.println(b1.i);
            System.out.println(b1.j);
        }
    }
19) class C
    {
        int i;
        void test1()
        {
            System.out.println("from test1");
        }
    }
    class D extends C
    {
        int j;
        void test2()
        {
            System.out.println("from test2");
        }
        public static void main(String[] args)
        {
            D obj=new D();
            obj.test1();
            obj.test2();
            System.out.println("-----");
            System.out.println(obj.i);
            System.out.println(obj.j);
        }
    }
20) class I
    {
        I()
        {
            System.out.println("I()");
        }
    }
    class J extends I
    {
        J()
        {
            System.out.println("J()");
        }
    }
    class S
    {
        public static void main(String[] args)
        {
            I i1=new I();
            System.out.println("-----");
            J j1=new J();
        }
    }

```

```

}
21) class E
{
    E()
    {
        System.out.println("E()");
        System.out.println("-----");
    }
    E(int i)
    {
        System.out.println("E(int)");
    }
}
class F extends E
{
    F()
    {
        System.out.println("F()");
    }
    F(int i)
    {
        System.out.println("F(int)");
    }
    public static void main(String[] args)
    {
        E e1=new E(10);
        System.out.println("-----");
        F obj=new F(10);
    }
}
22) class M
{
    M(int i)
    {
        System.out.println("int()");
    }
}
class N
{
    N(int i,int j)
    {
        System.out.println("int(), int()");
    }
    public static void main(String[] args)
    {
        N obj=new N(10,20);
        System.out.println("Success");
    }
}
23) class P
{
    P(int i)
    {
        System.out.println("P(int)");
    }
}
class Q extends P
{
    Q()
    {
        System.out.println("Q()");
    }
    public static void main(String[] args)
    {
        Q q=new Q();
    }
    static
    {
        System.out.println("From sib");
    }
    {
        System.out.println("from iib");
    }
}
24) class V
{
    V(int i)
    {

```

```

        System.out.println("V(int)");
    }
}
class W extends V
{
    W(int i)
    {
        System.out.println("W(int)");
    }
    public static void main(String[] args)
    {
        V v1=new V(10);
        System.out.println("-----");
        W w1=new W(20);
        System.out.println("-----");
    }
}
25) class G
{
    G(int i)
    {
        System.out.println("G(int)");
    }
}
class S extends G
{
    S()
    {
        super(10);
        System.out.println("S(int)");
    }
    public static void main(String[] args)
    {
        S obj=new S();
    }
}
26) class J
{
    J(int i,int j)
    {
        System.out.println("success");
    }
}
class S extends J
{
    S(int i, int j)
    {
        System.out.println("Very success");
    }
    public static void main(String[] args)
    {
        S obj=new S(10,20);
    }
}
27) class L
{
    {
        System.out.println("IIB1");
    }
    {
        System.out.println("IIB2");
    }
    static
    {
        System.out.println("SIB1");
    }
    static
    {
        System.out.println("SIB2");
    }
}
L()
{
    System.out.println("constructor");
}

```

```

    }
    L(int i)
    {
        System.out.println("L(int");
    }
    L(int i,int j)
    {
        System.out.println("L(int,int)");
    }
}
class S extends L
{
    S()
    {
        System.out.println("Constructor of S
class");
    }
    public static void main(String[] args)
    {
        S obj=new S();
    }
}
28) abstract class F
{
    abstract void test1();
    void test2()
    {
        System.out.println("from test1");
    }
}
class G extends F
{
    void test1()
    {
        System.out.println("from test2");
    }
    public static void main(String[] args)
    {
        G obj=new G();
        obj.test1();
        obj.test2();
        System.out.println("Done");
    }
}
abstract class H
{
    abstract void test1();
}
class I extends H
{
    void test1()
    {
        System.out.println("from test1");
    }
    public static void main(String[] args)
    {
        H h1=new H();
        System.out.println("Done");
    }
}
29) abstract class J
{
    abstract void test1();
    abstract void test2();
    void test3()
    {
        System.out.println("from test3");
    }
}
abstract class K extends J
{
    void test1()

```

```

    {
        System.out.println("from test1");
    }
}
class L extends K
{
    void test2()
    {
        System.out.println("from test2");
    }
    public static void main(String[] args)
    {
        L obj1=new L();
        obj1.test1();
        obj1.test2();
        obj1.test3();
        System.out.println("Done");
    }
}
30) abstract class K
{
    abstract void test1();
    abstract void test2();
    abstract void test3();
}
abstract class L extends K
{
    void test1()
    {
        System.out.println("from test1");
    }
    abstract void test4();
}
class M extends L
{
    void test2()
    {
        System.out.println("from test2");
    }
    void test3()
    {
        System.out.println("from test3");
    }
    void test4()
    {
        System.out.println("from test4");
    }
    public static void main(String[] args)
    {
        M m1=new M();
        m1.test1();
        m1.test2();
        m1.test3();
        m1.test4();
        System.out.println("Done");
    }
}
31) abstract class N
{
    void test1()
    {
        System.out.println("from test1");
    }
}
class O extends N
{
    public static void main(String[] args)
    {
        System.out.println("-----");
        O o1=new O();
        o1.test1();
    }
}

```

```

        System.out.println("-----");
    }
}
32) abstract class P
{
    void test1()
    {
        System.out.println("from test1");
    }
}
abstract class Q
{ }
class R extends P
{ }
class S extends Q
{ }
class T
{
    public static void main(String[] args)
    {
        System.out.println("-----!");
        System.out.println("-----!");
        R r1=new R();
        r1.test1();
        System.out.println("-----!");
        S s1=new S();
        System.out.println("-----!");
    }
}
33) abstract class U
{
    U()
    {
        System.out.println("hello world");
    }
    public static void main(String args[])
    {
        System.out.println("Success");
    }
}
34) abstract class V
{
    V()
    {
        System.out.println("V()");
    }
}
class W extends V
{
    W()
    {
        System.out.println("W()");
    }
    public static void main(String[] args)
    {
        W w1=new W();
        System.out.println("Done");
    }
}
35) abstract class X
{
    X(int i)
    {
        System.out.println("X(int)");
    }
    abstract void test1();
}
abstract class Y extends X
{
    Y()
    {
        super(90);
        System.out.println("Y()");
    }
}

```

```

class Z extends Y
{
    Z()
    {System.out.println("Z()");
    }
    void test1()
    {System.out.println("from test1");
    }
    public static void main(String[] args)
    {
        Z z1=new Z();
        System.out.println("-----");
        z1.test1();
    }
}
36) class A
{
    abstract void test1();
}
class B extends A
{
    void test1()
    {
        System.out.println("from test1");
    }
    public static void main(String[] args)
    {
        B b=new B();
        System.out.println("Done");
    }
}
37) abstract class C
{
    abstract void test1();
}
class D extends C
{
    void test1()
    {
        System.out.println("from test1");
    }
    public static void main(String[] args)
    {
        C c1=new C();
        System.out.println("Done");
    }
}
38) package com;
class A
{
    static int i;
    public static void main(String[] args)
    {
        System.out.println("From com.A");
        System.out.println(com.A.i);
    }
}
39) package pack1;
class B
{
    private int i=10;
    private String s1="sdjinfosoft";
    public static void main(String[] args)
    {
        B b1=new B();
        B b2=new B();
        System.out.println(b1.i);
        System.out.println(b2.s1);
    }
}
40) package pack1;
class C
{
    private int i;
}

```

```

    }
    public class D
    {
        public static void main(String[] args)
        {
            C c1=new C();
            System.out.println(c1.i);
        }
    }
41) package pack1;
    class D
    {
        private int j;
        private int test()
        {
            System.out.println("From test");
            System.out.println(j);
            return j;
        }
    }
    class E
    {
        public static void main(String[] args)
        {
            D d1=new D();
            System.out.println(d1.test());
            System.out.println("from main");
        }
    }
42) package pack1;
    class F
    {
        private static int i=90;
        static int j=test();
        private static int test()
        {
            System.out.println("from private test:"+ i );
            return i;
        }
    }
    class G
    { public static void main(String[] args)
      {
        System.out.println(F.j);
      }
    }
43) package pack1;
    class H
    { private int i;
      void set(int k, int j)
      {
        j=k;
        i=j;
      }
      int get()
      {
        return i;
      }
    }
    class I
    {
        public static void main(String[] args)
        {
            H h1=new H();
            System.out.println(h1.get());
            h1.set(90,50);
            System.out.println(h1.get());
        }
    }
44) package pack1;
    class K
    {

```

```

protected K()
{
    System.out.println("K()");
}
private K(int i)
{
    System.out.println("K(int)");
}
}
class L
{
    public static void main(String[] args)
    {
        K k1=new K();
        System.out.println("Done");
    }
}
45) package pack1;
    class Q
    {
        private Q()
        {
            System.out.println("Q()");
        }
        public Q(int i)
        {
            this();
            System.out.println("Q(int)");
        }
    }
    class R extends Q
    {
        R()
        {
            super(10);
        }
        public static void main(String[] args)
        {
            R r1=new R();
            System.out.println("Done");
        }
    }
46) interface A
    {
        void test();
    }
    public class B implements A
    {
        public void test()
        {
            System.out.println("From test");
        }
        public static void main(String[] args)
        {
            B b1=new B();
            b1.test();
        }
    }
47) interface B
    {
        static void test();
    }
    public class D implements A
    {
        public void test()
        {
            System.out.println("from D test()");
        }
    }
48) interface E
    {
        void test1();
    }

```

```

    }
    public class F implements E
    {
        void test1()
        {
            System.out.println("from F test()");
        }
    }
49) interface A
    {
        void getName();
    }
    interface B
    {
        void getAge();
    }
    public class H implements A,B
    {
        public void getName()
        {
            System.out.println("From A");
        }
        public void getAge()
        {
            System.out.println("From B");
        }
        public static void main(String args[])
        {
            H h=new H();
            h.getName();
            h.getAge();
        }
    }
50) interface G
    {
        void getName();
    }
    interface G1
    {
        void getAge();
    }
    public class H implements G,G1
    {
        public void getName()
        {
            System.out.println("From A");
        }
    }
51) interface K
    {
        int i;
    }
    public class H implements K
    {
        public static void main(String[] args)
        {
            H h1=new H();
            System.out.println(h1.i);
        }
    }
52) interface L
    {
        L()
        {
            System.out.println("L()");
        }
    }
    public class M implements L
    {
        M()
        {
            System.out.println("M()");
        }
    }

```

```

    }
    public static void main(String[] args)
    {
        M m1=new M();
        System.out.println("from Main");
    }
}
53) interface L
    {
        void on();
    }
    class Do
    {
        void off()
        {
            System.out.println("from Do ");
        }
    }
    public class Fan extends Do implements L
    {
        public void on()
        {
            System.out.println("from interface L");
        }
        public static void main(String[] args)
        {
            System.out.println("from Main");
            Fan f1=new Fan();
            f1.on();
            f1.off();
        }
    }
54) interface Super
    {
        int super1();
    }
    class N
    {
        void test1()
        {
            System.out.println("from N");
        }
    }
    public class Sub extends N,Super
    {
        public int Super1()
        {
            System.out.println("from Supper");
        }
    }
55) interface Yess
    {
        static
        { System.out.println("yess SIB");
        }
    }
    public class Ok implements Yess
    {
        static
        {System.out.println("Ok SIB");
        }
        public static void main(String[] args)
        { System.out.println("form Ok main");
        }
    }
56) interface SDJ
    {void Corjava();
    }
    interface Tech extends SDJ
    {
        void J2EE();
    }
    public class Extend implements Tech

```

```

{   public void Corjava()
    {       System.out.println("from Core Java");
    }
    public void J2EE()
    {       System.out.println("from J2EE");
    }
    public static void main(String[] args)
    {       Extend cls=new Extend();
            cls.Corjava();
            cls.J2EE();
    }
}

```

57) interface SDJINFOSOFT

```

{   void morning();
    void afternoon();
}
abstract class Session implements SDJINFOSOFT
{   public void morning()
    {       System.out.println("form morning");
    }
    abstract void evening();
}
public class Y extends Session implements SDJINFOSOFT
{
    public void afternoon()
    {       System.out.println("form afternoon ");
    }
    public void evening()
    {
        System.out.println("from evening");
    }
    public static void main(String[] args)
    {
        System.out.println("from main");
        Y y1=new Y();
        y1.morning();
        y1.afternoon();
        y1.evening();
    }
}

```

58) abstract class First

```

{   abstract void firstTest();
}
interface Second
{
    abstract void firstTest();
}
public class Last implements Second extends First
{
    public void firstTest()
    {
        System.out.println("firstTest()");
    }
    public static void main(String[] args)
    {
        System.out.println("done");
    }
}

```