1. Describe non-static global members.
   Non-static global members can't be used inside a static context without a reference variable.
   To use non-static global members in a static context a reference variable is required.
   Non-static global members are loaded in memory while creating an object of a class.
   Non-static global members are loading into memory only once for an object.
   In order, to access non-static members from the object maximum 1 reference variable is required, using reference variable non-static global variable can be accessed.

2. What is a reference variable?
   All classes, interface,enum types are considered as derived datatypes. A variable with its datatype as derived datatype is called as a reference variable.A reference variable
   can be created as follow
   derivedDatatype refVar = new Constructor;
   (object creation to refVar(reference Variable))

3. How many derived datatypes are there in java?
   Enormous. All classes, interfaces, enum defined within the lib as well as created by programmer.

4. Different ways to develop a multiple classes.
   1. Developing separate independent java files for each independent classes.
   2. Developing multiple classes in one java file
   3. Developing separate classes in separate java files and using in each other.

5. Can a static keyword be used with members within block?
   No. ‘static’ keyword can be used only with members of the class.
   On using static within a block it results into compile time error.

6. Where does object, local variable, static members,reference variable get created in memory?
   Objects and static members are created on heap.
   Local variables and reference variable are created on stack.

7. To refer non-static members associated with a specific object. Write a syntax.
   derivedDatatype referenceVar = new Constructor;
   referenceVar.nonStaticMemberName;

8. For an object, how many reference variable can be there ?
   An object can have any no. of reference variables.

9. Can an object be referred from multiple methods?
   Yes.If methods have a reference variable to object.

10. Is it possible to reach object from multiple methods?
    Yes, if there are reference variable from multiple methods.

11. How many objects can be referred by reference variable at any point of time?
    A reference variable can refer maximum one object at a time.

12. Define: Live Object.
    A object with atleast one reference variable is called as a live object.

13. Define: Abandoned Object.
    A object which doesn’t have minimum 1 reference variable is called as an abandoned object.

14. What type of object can a reference variable refer to?
    A reference variable can refer to an object of same derivedDatatype as the one for reference variable.

15. What happens if non static member is being referred in static context without reference variable?
    Compile time error. Stating as:
    non-static [member] ___ cannot be referenced from static context.
    [member]→ method, variable → name of method or variable.

16. Define:Pass-by-value.
    While calling a method modification in target method is not affecting source it is pass-by-value.
    Primitive datatype variables will be passed by value.

17. Define:Pass-by-reference.
    While calling a method modification in target method is affecting source it is pass-by-reference.
    Derived datatype variables will be passed by reference.

18. What is a constructor?

A constructor is a non-static initialization block with same name as a class name and no return type.

19. How many constructors can a class have?
    A class can have any number of constructors. But each constructor should have different signatures.

20. What do you mean by constructor overloading?
    Incorporating multiple constructors in a same class with different signature is called as constructor overloading.

21. Can a object be created using any constructor?
    No.To create an object of a class you need to specify one of the available constructors for the class.

22. When does a compiler provide default no-arg constructor?
    Compiler provides no-arg default constructor only when a class does not contain any constructor.

23. By default, how many constructors get executed while object is being created?
    Only one.

24. Is multiple constructor execution possible for an object of a java class?
    Yes. To execute more than one constructor use this statement in the constructor body.

25. Specify difference between constructor and method.
    Method has a return type.
    Constructor doesn’t have a return type.

26. Why constructor overloading is required?

To provide multiple ways for creating an object.

27. Is recursion possible for constructors and methods in java?
    In case of constructor if there is recursive calling code then it will result in compile-time error.
    In case of method, if there is recursive calling code then compilation will be successful but it will result in runtime error.

28. What is the difference between SIB and constructor?
    SIB is executed while loading class into memory.
    Constructor executes while object is being created.

29. What is the purpose of Instance Initialization Block (IIB)?
    IIB is meant for code re-usability purpose.

30. How many times does IIB get executed when multiple constructors are executed for object creation?
    IIB is executing only once for an object though multiple constructors are executing for one object creation.
    IIB execution is object-wise and not constructor-wise.

31. Usage of global variable inside initializer with forward reference causes Illegal Forward Reference

32. Is usage of method inside initializer with forward reference possible?
    Yes. It is possible.

33. Is it possible to develop an empty .java file? Can it run?

Yes. It is possible. On compilation, it won't generate .class file since no class is available. Since .class file is not generated it can't run.

34. What are the different types of members a java file can contain?

A java file can have any number of enum, classes, annotations and interfaces in a java file.

---

1. Program

```
class A
{
    int i;
    public static void main(String[] args)
    {
        System.out.println(i);
    }
}
```

Output→ Compile time error

---

2. Program

```
class B
{
    void test()
    {
    }
    public static void main(String[] args)
    {
        test();
    }
}
```

Output→ Compile time error

3.Program

```
class C
{
    int i;
    static void test2()
    {
        i = 10;
    }
}
```

Output→ Compile time error

4.Program

```
class D
{
    static void test1()
    {
        test2();
    }
    void test2()
    {

    }
}
```

Output→ Compile time error

5.Program

```
class E
{
    int i = 20;

    static
    {
        i = 10;
    }
}
```
Output→ Compile time error

6.Program

```
class F
{
    void test1()
    {
    }

    static
    {
        test1();
    }
}
```
Output→ Compile time error

7.Program

```
class G
{
    int i;
    public static void main(String[] args)
    {
        G obj = new G();
```

```
        System.out.println(obj.i);
    }
}
Output→ 0
```

8.Program

```
class H
{
    int i;
    public static void main(String[] args)
    {
        H h1 = new H();
        System.out.println(h1.i);
        System.out.println(h1.i);
        System.out.println(h1.i);
    }
}
Output→ 0
        0
        0
```

9.Program

```
class I
{
    void test1()
    {
        System.out.println("from test1()");
    }
    public static void main(String[] args)
    {
        System.out.println("main begin");
```

```
            I obj = new I();
            obj.test1();
            System.out.println("main end");
        }
    }
    Output→ main begin
            from test1()
            main end
```

10.Program

```
    class J
    {
        void test1()
        {
            System.out.println("test1");
        }
        public static void main(String[] args)
        {
            J j1 = new J();
            System.out.println("main begin");
            j1.test1();
            System.out.println("-----");
            j1.test1();
            System.out.println("-----");
            System.out.println("main end");
        }
    }
Output→ main begin
        test1
        -----
        test1
        -----
         main end
```

11.Program

```
class K
{
    void test1()
    {
        System.out.println("from test1");
    }
    void test2()
    {
        System.out.println("from test2");
    }
    public static void main(String[] args)
    {
        System.out.println("main begin");
        K k1 = new K();
        k1.test1();
        System.out.println("---------");
        k1.test2();
        System.out.println("---------");
        k1.test1();
        System.out.println("---------");
        k1.test2();
        System.out.println("---------");
        System.out.println("main end");
    }
}
```

Output→
```
main begin
from test1
---------
from test2
---------
from test1
---------
from test2
```

```
---------
main end
```

12.Program

```
class L
{
    int i;
    void test1()
    {
        System.out.println("from test1");
    }
    public static void main(String[] args)
    {
        System.out.println("main begin");
        L obj = new L();
        obj.test1();
        System.out.println(obj.i);
        System.out.println("main end");
    }
}
```

```
Output→ main begin
        from test1
        0
        main end
```

13.Program

```
class M
{
    int i, j, m;
    public static void main(String[] args)
    {
        System.out.println("main begin");
```

```
            M obj = new M();
            System.out.println(obj.i);
            System.out.println(obj.j);
            System.out.println(obj.m);
            System.out.println("main end");
        }
    }
Output→ main begin
        0
        0
        0
        main end
```

14.Program

```
class N
{
    int i;
    static void test1()
    {
        N n1 = new N();
        n1.i = 30;
    }
}
Output→Run time error
```

15.Program

```
class P
{
    int i;
    static void test1()
    {
        P obj = new P();
```

```
        System.out.println(obj.i);
    }
}
```
Output→ Run time error

16.Program

```
class Q
{
    int i;

    static
    {
        Q q1 = new Q();
        q1.i = 20;
    }
}
```
Output→ Run time error

17.Program

```
class R
{
    static
    {
        R r1 = new R();
        r1.test1();
    }

    void test1()
    {
        System.out.println("Hello World!");
    }
}
```

Output→ Run time error

18.Program

```
class S
{
    int i;
    public static void main(String[] args)
    {
        S s1 = new S();
        System.out.println(s1.i);
        s1.i = 20;
        System.out.println(s1.i);
    }
}
```

Output→ 0
20

19.Program

```
class T
{
    int i;
    static void test()
    {
        obj.i = 10;
    }
    public static void main(String[] args)
    {
        T obj = new T();
        test();
        System.out.println(obj.i);
    }
}
```

Output→ Compile time error

20.Program

```
class U
{
    int i;
    static void test()
    {
        U obj = new U();
        obj.i = 20;
    }
    public static void main(String[] args)
    {
        System.out.println(obj.i);
    }
}
```

Output→ Compile time error

21.Program

```
class V
{
    int i;
    public static void main(String[] args)
    {
        V v1 = new V();
        V v2 = v1;
        v1.i = 10;
        System.out.println(v1.i);
        System.out.println(v2.i);
        System.out.println("-----");
        v2.i = 20;
        System.out.println(v1.i);
```

```
            System.out.println(v2.i);
            System.out.println("-----");
        }
    }
Output→ 10
        10
        -----
        20
        20
        -----
```

22.Program

```
    class W
    {
        double j;
        public static void main(String[] args)
        {
            W obj1 = new W();
            System.out.println(obj1.j);
            W obj2 = obj1;
            obj2.j = 5.0;
            System.out.println(obj1.j);
            System.out.println(obj2.j);
            W obj3 = obj2;
            obj2.j = 6.0;
            System.out.println(obj3.j);
            System.out.println(obj2.j);
            System.out.println(obj1.j);

        }
    }

Output→ 0.0
        5.0
```

```
    5.0
    6.0
    6.0
    6.0
```

23.Program

```
class X
{
    int i;
    public static void main(String[] args)
    {
        X x1 = new X();
        X x2 = new X();
        x1.i = 10;
        x2.i = 20;
        System.out.println(x1.i);
        System.out.println(x2.i);
    }
}
Output→ 10
        20
```

24.Program

```
class Y
{
    int i;
    public static void main(String[] args)
    {
        Y y1 = new Y();
        System.out.println(y1.i);
        Y y2 = new Y();
        System.out.println(y2.i);
```

```
            Y y3 = new Y();
            System.out.println(y3.i);
            y1.i = 2;
            y2.i = 3;
            y3.i = 4;
            System.out.println(y1.i);
            System.out.println(y2.i);
            System.out.println(y3.i);

        }
    }
Output→ 0
        0
        0
        2
        3
        4
```

25.Program

```
    class Z
    {
        int i;
        public static void main(String[] args)
        {
            Z z1 = new Z();
            Z z2 = z1;
            Z z3 = new Z();
            Z z4 = z3;
            z1.i = 10;
            z4.i = 20;
            System.out.println(z1.i);
            System.out.println(z2.i);
            System.out.println(z3.i);
            System.out.println(z4.i);
```

```
        }
    }
Output→ 10
        10
        20
        20
```

26.Program

```
class A
{
    public static void main(String[] args)
    {
        A a1 = new A();
        A a2 = a1;
        System.out.println(a1);
        System.out.println(a2);
    }
}
Output→ A@19e0bfd
        A@19e0bfd
```

---

27. Program

```
class B
{
    public static void main(String[] args)
    {
        B b1 = new B();
        B b2 = new B();
        B b3 = b2;
        B b4 = b1;
        System.out.println(b1);
        System.out.println(b2);
        System.out.println(b3);
```

```
            System.out.println(b4);
        }
    }
Output→ B@19e0bfd
        B@139a55
        B@139a55
        B@19e0bfd
```

---

28. Program

```
class C
{
    void test()
    {
        System.out.println("test:" + this);
    }
    public static void main(String[] args)
    {
        C c1 = new C();
        System.out.println("main:" + c1);
        c1.test();
    }
}
Output→ C@19e0bfd
        C@19e0bfd
```

---

29. Program

```
class D
{
    int i;
    void test()
    {
        System.out.println(this.i);
    }
    public static void main(String[] args)
```

```
    {
        D d1 = new D();
        d1.i = 10;
        d1.test();
    }
}
Output→ 10
```

---

30. Program

```
class E
{
    int i;
    void test()
    {
        System.out.println(this.i);
        this.i = 110;
    }
    public static void main(String[] args)
    {
        E e1 = new E();
        System.out.println(e1.i);
        e1.test();
        System.out.println(e1.i);
    }
}
Output→ 0
        0
        110
```

---

31. Program

```
class F
{
    public static void main(String[] args)
    {
        System.out.println(this);
    }
}
```
Output→ Compile time error

---

32. Program

```
class G
{
    static void test()
    {
        System.out.println(this);
    }
}
```
Output→ Compile time error

---

33. Program

```
class H
{
    static
    {
        System.out.println(this);
    }
}
```
Output→ Compile time error

---

34. Program

```
class I
{
    int x;
    void test()
    {
        System.out.println(x);
    }
    public static void main(String[] args)
    {
        System.out.println(x);
    }
}
```
Output→ Compile time error

---

35. Program

```
class J
{
    int x;
    void test()
    {
        System.out.println(x);
    }
    public static void main(String[] args)
    {
        J obj = new J();
        System.out.println(obj.x);
        obj.test();
    }
}
```
Output→ 0
0

---

36. Program

```
class K
{
    int i;
    void test1()
    {
        i = 10;
    }
    public static void main(String[] args)
    {
        K k1 = new K();
        System.out.println("a:" + k1.i);
        k1.test1();
        System.out.println("b:" + k1.i);
    }
}
Output→ a:0
        b:10
```

---

37. Program

```
class L
{
    int i;
    void test1()
    {
        i = 20;
    }
    public static void main(String[] args)
    {
        L obj1 = new L();
        System.out.println("a:" + obj1.i);
        obj1.test1();
        System.out.println("b:" + obj1.i);
        L obj2 = new L();
        System.out.println("c:" + obj2.i);
        obj2.test1();
```

```
        System.out.println("d:" + obj2.i);
    }
}
Output→ a:0
        b:20
        c:0
        d:20
```

---

38. Program

```
class M
{
    int i;
    void test1()
    {
        System.out.println("a:" + i);
        test2();
        System.out.println("b:" + i);
    }
    void test2()
    {
        i = 30;
    }
    public static void main(String[] args)
    {
        M m1 = new M();
        System.out.println("c:" + m1.i);
        m1.test1();
        System.out.println("d:" + m1.i);
        m1.i = 10;
        m1.test1();
        System.out.println("e:" + m1.i);
        m1.test2();
        System.out.println("f:" + m1.i);
    }
}
```

```
Output→ c:0
        a:0
        b:30
        d:30
        a:10
        b:30
        e:30
        f:30
```

---

39. Program

```
class N
{
    int i;
    static void test1(N obj)
    {
        obj.i = 30;
    }
    public static void main(String[] args)
    {
        N n1 = new N();
        n1.i = 1;
        System.out.println("a:" + n1.i);
        test1(n1);
        System.out.println("b:" + n1.i);
    }
}
Output→ a:1
        b:30
```

---

40. Program

```
class P
{
    int i;
    void test1()
    {
        i = 1;
    }
    static void test2(P obj)
    {
        System.out.println("a:" + obj.i);
        obj.test1();
        System.out.println("b:" + obj.i);
    }
    void test3()
    {
        System.out.println("c:" + this.i);
        test1();
        System.out.println("d:" + i);
    }
    public static void main(String[] args)
    {
        P p1 = new P();
        p1.test1();
        System.out.println("e:" + p1.i);
        p1.i = 2;
        test2(p1);
        System.out.println("f:" + p1.i);
        p1.i = 3;
        p1.test3();
        System.out.println("g:" + p1.i);
    }
}
```

Output→ e:1
a:2

```
    b:1
    f:1
    c:3
    d:1
    g:1
```

---

41. Program

```
class Q
{
    int i;
    void test()
    {
        Q q1 = new Q();
        q1.i = i;
        System.out.println("test-a:" + q1.i);
        System.out.println("test-b:" + i);
    }
    public static void main(String[] args)
    {
        Q obj = new Q();
        obj.i = 20;
        obj.test();
    }
}
Output→  test-a:20
         test-b:20
```

---

42. Program

```
class R
{
    int i;
    static void test(R r1)
    {
        R r2 = new R();
        r2.i = r1.i;
```

```
        System.out.println("test-a:" + r1.i);
        System.out.println("test-b:" + r2.i);
    }
    public static void main(String[] args)
    {
        R obj = new R();
        obj.i = 70;
        test(obj);
    }
}
Output→  test-a:70
         test-b:70
```

---

43. Program

```
class S
{
    int i;
    static void test(S s1)
    {
        s1.i = 10;
    }
    public static void main(String[] args)
    {
        S obj = new S();
        obj.i = 5;
        System.out.println("a:" + obj.i);
        test(obj);
        System.out.println("b:" + obj.i);
    }
}
Output→  a:5
         b:10
```

---

44. Program

```
class T
{
    int i;
    static void test1(T t1, T t2)
    {
        int x = t1.i;
        t1.i = t2.i;
        t2.i = x;
    }
    void test2(T t1)
    {
        int x = t1.i;
        t1.i = this.i;
        this.i = x;
    }
    public static void main(String[] args)
    {
        T obj1 = new T(), obj2 = new T();
        obj1.i = 1;
        obj2.i = 2;
        test1(obj1, obj2);
     System.out.println(obj1.i + "," + obj2.i);
        obj1.test2(obj2);
     System.out.println(obj1.i + "," + obj2.i);
    }
}
Output→  2,1
         1,2
```

---

45. Program

```
class A
{
    int i = 10;
    public static void main(String[] args)
    {
```

```
        A a1 = new A();
        System.out.println(a1.i);
    }
}
```
Output→ 10

---

46. Program

```
class B
{
    int i = test();
    static int test()
    {
        System.out.println("from test:");
        return 10;
    }
    public static void main(String[] args)
    {
        System.out.println("main begin ");
        B b1 = new B();
        System.out.println(b1.i);
        System.out.println("main end ");
    }
}
```
Output→ main begin
from test:
10
main end

---

47. Program

```
class C
{
    int i = test();
    int test()
    {
        System.out.println("from test:" + i);
        return 10;
    }
    public static void main(String[] args)
    {
        System.out.println("main begin");
        C c1 = new C();
        System.out.println(c1.i);
        System.out.println("main end");
    }
}
```

Output→ main begin
from test:0
10
main end

---

48. Program

```
class D
{
    int i = test1();
    int j = test1();
    int test1()
    {
        System.out.print("from test1:");
        System.out.print("i = " + i);
        System.out.println(",j = " + j);
        return 10;
    }
    public static void main(String[] args)
    {
```

```
        System.out.println("main begin");
        D d1 = new D();
        System.out.println("----------");
        System.out.print(d1.i);
        System.out.println("," + d1.j);
        System.out.println("main end");
    }
}
Output→  main begin
         from test1:i = 0,j = 0
         from test1:i = 10,j = 0
         ----------
         10,10
         main end
```

---

49. Program

```
class E
{
    int i;
    E()
    {
        System.out.println("E():" + i);
        i = 20;
    }
    public static void main(String[] args)
    {
        System.out.println("main begin");
        E e1 = new E();
        System.out.println(e1.i);
    }
}
Output→  main begin
         E():0
         20
```

---

50. Program

```
class F
{
    int i;
    int j;
    F()
    {
        System.out.print("F():");
        System.out.print("i = " + i);
        System.out.println(", j = " + j);
        i = 10;
        j = 20;
    }
    public static void main(String[] args)
    {
        System.out.println("F-main begin");
        F f1 = new F();
        System.out.print("i = " + f1.i);
        System.out.println(", j = " + f1.j);
        System.out.println("F-main end");
    }
}
```

Output→

```
F-main begin
F():i = 0, j = 0
i = 10, j = 20
F-main end
```

---

51. Program

```
class G
{
    int i = test1();
    G()
    {
        System.out.println("G():" + i);
```

```
        i = 20;
    }
    int test1()
    {
        System.out.println("test1:" + i);
        return 30;
    }
    public static void main(String[] args)
    {
        System.out.println("main begin");
        G g1 = new G();
        System.out.println("main:" + g1.i);
        System.out.println("main end");
    }
}
```

Output→ main begin
test1:0
G():30
main:20
main end

---

52. Program

```
class H
{
    int x = test1();

    H()
    {
        System.out.println("H():" + x);
        x = 20;
    }

    H()
    {
        System.out.println("H():" + x);
```

```
        x = 40;
    }

    int test1()
    {
        System.out.println("test1:" + x);
        return 10;
    }

    public static void main(String[] args)
    {
        System.out.println("main begin");
        H h1 =new H();
        System.out.println("main end");
    }
}
```

Output→ Compile time error

---

53. Program

```
class I
{
    static int x = test1();
    int y = test2();

    static int test1()
    {
        System.out.println("from test1:" + x);
        return 10;
    }

    int test2()
    {
        System.out.println("from test2:" + y);
        return 20;
    }
```

```
    I()
    {
        System.out.println("I():" + y);
        y = 100;
    }

    static
    {
        System.out.println("SIB:" + x);
        x = 200;
    }

    public static void main(String[] args)
    {
        System.out.println("main begin");
        System.out.println("main:" + x);
        I obj = new I();
        System.out.println("main:" + obj.y);
        System.out.println("main end");
    }
}
```

```
Output→  from test1:0
         SIB:10
         main begin
         main:200
         from test2:0
         I():20
         main:100
         main end
```

54. Program

```
class J
{
    static int x;
    public static void main(String[] args)
    {
        J obj = new J();
        System.out.println(x);
        System.out.println(J.x);
        System.out.println(obj.x);
    }
}
```

Output→ 0
0
0

---

55. Program

```
class K
{
    static int x;
    void test()
    {
        x = 10;
    }
    public static void main(String[] args)
    {
        K k1 = new K();
        k1.test();
        System.out.println(x);
    }
}
```

Output→ 10

---

56. Program

```
class L
{
    static int count;
    L()
    {
        count ++;
    }
    public static void main(String[] args)
    {
        L obj1 = new L();
        L obj2 = new L();
        L obj3 = new L();
        L obj4 = new L();
        System.out.println(count);
    }
}
```
Output→ 4

---

57. Program

```
class M
{
    static int count;
    M()
    {
        count ++;
    }
    public static void main(String[] args)
    {
        M m1 = new M();
        M m2 = new M();
        M m3 = new M();
        M m4 = new M();
        System.out.println(m1.count);
        System.out.println(m2.count);
```

```
        System.out.println(m3.count);
        System.out.println(m4.count);
    }
}
Output→  4
         4
         4
         4
```

---

58. Program

```
class N
{
    static int count;
    N()
    {
        count ++;
    }
    public static void main(String[] args)
    {
        N n1 = new N();
        System.out.println(n1.count);
        N n2 = new N();
        System.out.println(n2.count);
        N n3 = new N();
        System.out.println(n3.count);
        N n4 = new N();
        System.out.println(n4.count);
        System.out.println(n1.count);
        System.out.println(n2.count);
        System.out.println(n3.count);
    }
}

Output→  1
         2
```

```
          3
          4
          4
          4
          4
```

---

59. Program

```
class O
{
    int i;
    O()
    {
        i++;
    }
    public static void main(String[] args)
    {
        O o1 = new O();
        O o2 = new O();
        O o3 = new O();
        O o4 = new O();
        System.out.println(o1.i);
        System.out.println(o2.i);
        System.out.println(o3.i);
        System.out.println(o4.i);
    }
}
Output→  1
         1
         1
         1
```

---

60. Program

```
class P
{
    P()
    {
        System.out.println("P()");
    }

    P(int i)
    {
        System.out.println("P(int)");
    }

    public static void main(String[] args)
    {
        P p1 = new P();
        System.out.println("-----");
        P p2 = new P(20);
        System.out.println("-----");
        P p3 = new P();
        System.out.println("-----");
        P p4 = new P(20);
        System.out.println("-----");
    }
}
Output→  P()
         -----
         P(int)
         -----
         P()
         -----
         P(int)
         -----
```

---

61. Program

```
class Q
{
    Q()
    {
        System.out.println("Q()");
    }
    Q(int i)
    {
        System.out.println("Q(int)");
    }
    Q(int i, int j)
    {
        System.out.println("Q(int, int)");
    }

    public static void main(String[] args)
    {
        Q q1 = new Q(10, 20);
        System.out.println("------");
        Q q2 = new Q();
        System.out.println("------");
        Q q3 = new Q(30);
        System.out.println("------");
        Q q4 = new Q();
        System.out.println("------");
    }
}
Output→  Q(int, int)
         ------
         Q()
         ------
         Q(int)
         ------
         Q()
      ------
```

---

62. Program

```
class R
{
    R(int i)
    {
        System.out.println("R(int i)");
    }
    R(int j)
    {
        System.out.println("R(int j)");
    }
    public static void main(String[] args)
    {
        R r1 = new R(9);
        System.out.println("done");
    }
}
```

Output→ Compile time error

---

63. Program

```
class S
{
    S(int i, double j)
    {

    }

    S(int m, double n)
    {

    }
}
```

Output→ Compile time error

---

64. Program

```
class T
{
    T(int i, double j)
    {
        System.out.println("T(int, double)");
    }
    T(double i, int j)
    {
        System.out.println("T(double, int)");
    }
    public static void main(String[] args)
    {
        T t1 = new T(2.4, 5);
        System.out.println("-------");
        T t2 = new T(10, 6.5);
    }
}
```

Output→ T(double, int)

T(int, double)

65. Program

```
class U
{
    static int count;
    U()
    {
        count ++;
        System.out.println("U()");
    }
    U(int i)
    {
```

```
        count ++;
        System.out.println("U(int)");
    }
    public static void main(String[] args)
    {
        U u1 = new U();
        System.out.println("-----");
        U u2 = new U(10);
        System.out.println("-----");
        U u3 = new U(20);
        System.out.println(count);
    }
}
Output→  U()
         -----
         U(int)
         -----
         U(int)
         3
```

---

66. Program

```
class V
{
    V(int i)
    {
        System.out.println("V(int)");
    }
    public static void main(String[] args)
    {
        V v1 = new V();
        System.out.println("done");
    }
}
Output→ Compile time error
```

---

67. Program

```
class W
{
    public static void main(String[] args)
    {
        W w1 = new W();
        System.out.println("done");
    }
}
```

Output→ done

---

68. Program

```
class X
{
    public static void main(String[] args)
    {
        X x1 = new X(90);
        System.out.println("done");
    }
}
```

Output→ Compile time error

---

69. Program

```
class Y
{
    Y(int i)
    {
        System.out.println("Y(int)");
    }
    public static void main(String[] args)
    {
        Y y1 = new Y();
        System.out.println("Hello World!");
```

```
    }
}
```
Output→ Compile time error

---

70. Program

```
class A
{
    A()
    {
        System.out.println("A()");
    }
    A(int i)
    {
        this();
        System.out.println("A(int)");
    }
    public static void main(String[] args)
    {
        A a1 = new A();
        System.out.println("------");
        A a2 = new A(20);
        System.out.println("------");
    }
}
```
Output→ A()

A()
A(int)

---

71. Program

```
class B
{
    B()
    {
        this(10);
        System.out.println("B()");
    }
    B(int i)
    {
        System.out.println("B(int)");
    }
    public static void main(String[] args)
    {
        B b1 = new B();
        System.out.println("-----");
        B b2 = new B(90);
    }
}
Output→  B(int)
         B()
         -----
         B(int)
```

---

72. Program

```
class C
{
    C()
    {
        this(2, 6);
        System.out.println("C()");
    }
    C(int i)
    {
        System.out.println("C(int)");
    }
```

```
    C(int i, int j)
    {
        System.out.println("C(int, int)");
    }
    public static void main(String[] args)
    {
        C c1 = new C();
        System.out.println("--------");
        C c2 = new C(20, 40);
        System.out.println("--------");
        C c3 = new C(100);
        System.out.println("--------");
    }
}
Output→  C(int, int)
         C()
         --------
         C(int, int)
         --------
         C(int)
         --------
```

---

73. Program

```
class D
{
    D()
    {
        System.out.println("D()");
        this(90);
    }
    D(int i)
    {
        System.out.println("D(int)");
    }
}
```

Output→ Compile time error

---

74. Program

```
class E
{
    E()
    {
        this(10);this(10);
        System.out.println("E()");
    }
    E(int i)
    {
        System.out.println("E(int)");
    }
}
```

Output→ Compile time error

---

75. Program

```
class F
{
    F()
    {
        this(10);
        System.out.println("F()");
    }
    F(int i)
    {
        this();
        System.out.println("F(int)");
    }
}
```

Output→ Compile time error

---

76. Program

```
class G
{
    G()
    {
        this();
        System.out.println("G()");
    }
}
```
Output→ Compile time error

---

77. Program

```
class H
{
    H()
    {
        this(2.4);
        System.out.println("H()");
    }
    H(String s1)
    {
        System.out.println("H(String)");
    }
}
```
Output→ Compile time error

---

78. Program

```
class I
{
    I()
    {
        System.out.println("I()");
    }

    {
```

```
        System.out.println("I-IIB");
    }

    public static void main(String[] args)
    {
        I obj1 = new I();
        System.out.println("-------");
        I obj2 = new I();
        System.out.println("-------");
    }
}
```

Output→
```
I-IIB
I()
-------
I-IIB
I()
-------
```

---

79. Program

```
class J
{
    J()
    {
        System.out.println("J()");
    }

    {
        System.out.println("J-IIB1");
    }

    public static void main(String[] args)
    {
        J j1 = new J();
        System.out.println("------");
        J j2 = new J();
```

```
        System.out.println("------");
    }

    {
        System.out.println("J-IIB2");
    }

}
Output→  J-IIB1
         J-IIB2
         J()
         ------
         J-IIB1
         J-IIB2
         J()
         ------
```

---

80. Program

```
class K
{
    K()
    {
        System.out.println("K()");
    }
    K(int i)
    {
        System.out.println("K(int)");
    }

    {
        System.out.println("K-IIB");
    }

    public static void main(String[] args)
    {
```

```
        K k1 = new K();
        System.out.println("-------");
        K k2 = new K(30);
        System.out.println("-------");
    }
}
Output→  K-IIB
         K()
         -------
         K-IIB
         K(int)
         -------
```

---

81. Program

```
class L
{
    L()
    {
        System.out.println("L()");
    }

    {
        System.out.println("L-IIB1");
    }

    L(int i)
    {
        System.out.println("L(int)");
    }

    {
        System.out.println("L-IIB2");
    }

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

```
    {
        L obj1 = new L();
        System.out.println("--------");
        L obj2 = new L(20);
        System.out.println("--------");
    }
}
Output→  L-IIB1
         L-IIB2
         L()
         --------
         L-IIB1
         L-IIB2
         L(int)
         --------
```

---

82. Program

```
class M
{
    {
        System.out.println("M-IIB");
    }

    M(int i)
    {
        System.out.println("M(int)");
    }

    M()
    {
        System.out.println("M()");
    }

    M(double i)
    {
```

```
        System.out.println("M(double)");
    }
    public static void main(String[] args)
    {
        M m1 = new M();
        System.out.println("-------");
        M m2 = new M(20);
        System.out.println("-------");
        M m3 = new M(2.0);
        System.out.println("-------");
    }
}
```

Output→

```
M-IIB
M()
-------
M-IIB
M(int)
-------
M-IIB
M(double)
-------
```

---

83. Program

```
class N
{
    N(int i)
    {
        this();
        System.out.println("N(int)");
    }

    {
        System.out.println("N-IIB");
    }
```

```
    N()
    {
        System.out.println("N()");
    }

    public static void main(String[] args)
    {
        N n1 = new N();
        System.out.println("-------");
        N n2 = new N(20);
        System.out.println("-------");
    }
}
```

Output→
```
N-IIB
N()
-------
N-IIB
N()
N(int)
-------
```

---

84. Program

```
class O
{
    O()
    {
        System.out.println("O()");
    }
    O(int i)
    {
        this();
        System.out.println("O(int)");
    }
    O(int i, int j)
    {
```

```
        this(i);
        System.out.println("O(int, int)");
    }

    {
        System.out.println("O-IIB1");
    }

    {
        System.out.println("O-IIB2");
    }

    public static void main(String[] args)
    {
        O x1 = new O();
        System.out.println("--------");
        O x2 = new O(4, 6);
        System.out.println("--------");
        O x3 = new O(4);
        System.out.println("--------");
    }
}
Output→  O-IIB1
         O-IIB2
         O()
         --------
         O-IIB1
         O-IIB2
         O()
         O(int)
         O(int, int)
         --------
         O-IIB1
         O-IIB2
       O()
       O(int)
```

----

85. Program

```
class P
{
    P()
    {
        System.out.println("P()");
    }
    P(int i)
    {
        this();
        System.out.println("P(int)");
    }

    static
    {
        System.out.println("P-SIB1");
    }

    {
        System.out.println("P-IIB1");
    }

    P(int i, int j)
    {
        this(j);
        System.out.println("P(int, int)");
    }

    {
        System.out.println("P-IIB2");
    }

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

```
    {
        System.out.println("main begin");
        P p1 = new P(90);
        System.out.println("--------");
        P p2 = new P(50, 60);
        System.out.println("--------");
        P p3 = new P();
        System.out.println("--------");
        System.out.println("main end");
    }
    static
    {
        System.out.println("P-SIB2");
    }
}
```

Output→

```
P-SIB1
P-SIB2
main begin
P-IIB1
P-IIB2
P()
P(int)
--------
P-IIB1
P-IIB2
P()
P(int)
P(int, int)
--------
P-IIB1
P-IIB2
P()
--------
main end
```

86. Program

```
class Q
{
    int i;
    Q(int i)
    {
        this.i = i;
        System.out.println("Q(int)");
    }
    public static void main(String[] args)
    {
        Q q1 = new Q(20);
        System.out.println(q1.i);
    }
}
Output→  Q(int)
         20
```

---

87. Program

```
class R
{
    int i;
    double j;
    R(int i, double j)
    {
        this.i = i;
        this.j = j;
    }
    public static void main(String[] args)
    {
        R r1 = new R(10, 5.8);
        System.out.println(r1.i);
        System.out.println(r1.j);
    }
}
Output→ 10
```

5.8

88. Program

```
class S
{
    int i;

    S(int i)
    {
        i = i;
    }
    public static void main(String[] args)
    {
        S s1 = new S(10);
        System.out.println(s1.i);
    }
}
```

Output→ 0

89. Program

```
class Person
{
    String name;
    int age;
    double weight;

    Person(String name,
           int age,
           double weight)
    {
        this.name = name;
        this.age = age;
        this.weight = weight;
    }
```

```
    void printInfo()
    {
        System.out.print(name + ", ");
        System.out.print(age + ", ");
        System.out.println(weight);
    }
}
class T
{
    public static void main(String[] args)
    {
    Person p1 = new Person("Vijay", 25, 55.90);
    Person p2 = new Person("Kiran", 35, 65.90);
    Person p3 = new Person("Ramu", 27, 56.90);
    p1.printInfo();
    p2.printInfo();
    p3.printInfo();
    }
}
Output→  Vijay, 25, 55.9
         Kiran, 35, 65.9
         Ramu, 27, 56.9
```

---

90. Program

```
class U
{
    int i;
}
class V
{
    int j;
    U obj;
}
class W
{
```

```
    public static void main(String[] args)
    {
        U u1 = new U();
        u1.i = 10;
        V v1 = new V();
        v1.j = 20;
        v1.obj = u1;
        System.out.println(v1.j);
        System.out.println(v1.obj.i);
    }
}
Output→  20
         10
```

---

91. Program

```
class X
{
    int i;
    int j;
}
class Y
{
    int m;
    int n;
    X x1;
    X x2;
}
class Z
{
    public static void main(String[] args)
    {
        X obj1 = new X();
        obj1.i = 10;
        obj1.j = 11;
```

```
        Y y1 = new Y();
        y1.m = 12;
        y1.n = 13;
        y1.x1 = obj1;
        y1.x2 = new X();
        y1.x2.i = 14;
        y1.x2.j = 15;
        System.out.println(y1.m);
        System.out.println(y1.n);
        System.out.println(y1.x1.i);
        System.out.println(y1.x1.j);
        System.out.println(y1.x2.i);
        System.out.println(y1.x2.j);
    }
}
Output→  12
         13
         10
         11
         14
         15
```

---

92. Program

```
class Address
{
    String houseNo;
    String streetName;
}
class Person
{
    String firstName;
    int age;
    double weight;
    Address permanentAddress;
    void printPersonInfo()
```

```
    {
System.out.print(firstName + ", ");
System.out.print(age + ", ");
System.out.print(weight + ", ");
System.out.print(permanentAddress.houseNo + ", ");
System.out.println(permanentAddress.streetName);
    }
}
class A
{
public static void main(String[] args)
 {
    Address a1 = new Address();
    a1.houseNo = "123/M";
    a1.streetName = "BTM";
    Person p1 = new Person();
    p1.firstName = "Vijay";
    p1.age = 22;
    p1.weight = 55.89;
    p1.permanentAddress = a1;
    Person p2 = new Person();
    p2.firstName = "Kiran";
    p2.age = 32;
    p2.weight = 65.89;
    p2.permanentAddress = new Address();
    p2.permanentAddress.houseNo = "304/S";
    p2.permanentAddress.streetName = "JP Nagar";

    p1.printPersonInfo();
    p2.printPersonInfo();
 }
}
```

```
Output→  Vijay, 22, 55.89, 123/M, BTM
         Kiran, 32, 65.89, 304/S, JP Nagar
```

93. Program

```
class B
{
    int i;
}
class C
{
    int j;
    B b1;
    C(int j, B b1)
    {
        this.j = j;
        this.b1 = b1;
    }
    void printInfo()
    {
        System.out.print(j + ", ");
        System.out.println(b1.i);
    }
}
class D
{
    public static void main(String[] args)
    {
        B b1 = new B();
        b1.i = 10;
        C c1 = new C(20, b1);
        B b2 = new B();
        b2.i = 40;
        C c2 = new C(30, b2);
        c1.printInfo();
        c2.printInfo();
    }
}
```

```
Output→  20, 10
         30, 40
```
