```
1. class Test {
   int x = 10;
   Test(int x) { this.x = x; }
   void show() { System.out.println(x); }
   public static void main(String[] args) {
      Test t = new Test(50);
      t.show();
   }
}

2. class A {
   static int count = 0;
   A(){ count++; }
   public static void main(String[] args){
      new A(); new A(); new A();
      System.out.println(count);
   }
}

3. class Demo {
   Demo(){ System.out.println("A"); }
   Demo(int a){ System.out.println("B"); }
   public static void main(String[] args){ new Demo(10); }
}

4. class Base {
   int x = 5;
}
class Derived extends Base {
   int x = 10;
   void show(){ System.out.println(x + " " + super.x); }
   public static void main(String[] args){ new
Derived().show(); }
```

```
}

5. class P {
   void show(){ System.out.println("P"); }
}
class Q extends P {
   void show(){ System.out.println("Q"); }
   public static void main(String[] args){
      P obj = new Q();
      obj.show();
   }
}

6. class A {
   final void display(){ System.out.println("A"); }
}
class B extends A {
   // display() override attempt
   public static void main(String[] args){ new B().display(); }
}

7. class X {
   void test(int a){ System.out.println("int"); }
   void test(double a){ System.out.println("double"); }
   public static void main(String[] args){ new X().test(5); }
}

8. class Parent {
   Parent(){ System.out.println("Parent"); }
}
class Child extends Parent {
   Child(){ System.out.println("Child"); }
   public static void main(String[] args){ new Child(); }
```

```
}

9. interface A { void show(); }
class B implements A {
   public void show(){ System.out.println("Hello"); }
   public static void main(String[] args){ A obj = new B();
obj.show(); }
}

10. class A {
   A(){ System.out.println("A"); }
}
class B extends A {
   B(){ System.out.println("B"); }
}
class C extends B {
   C(){ System.out.println("C"); }
   public static void main(String[] args){ new C(); }
}

11. class Test {
   void show(){ System.out.println("A"); }
}
class Demo extends Test {
   void show(){ System.out.println("B"); }
   void display(){ super.show(); }
   public static void main(String[] args){ new
Demo().display(); }
}

12. class A {
   static { System.out.println("Block"); }
```

```
    public static void main(String[] args)
{ System.out.println("Main"); }
}

13. class A {
    private void show(){ System.out.println("A"); }
    public static void main(String[] args){
        A obj = new A();
        obj.show();
    }
}

14. class Test {
    void m1(){ System.out.println("m1"); }
    void m1(int x){ System.out.println("m1-int"); }
    public static void main(String[] args){ new Test().m1(); }
}

15. class A {
    void call(){ System.out.println("A"); }
}
class B extends A {
    void call(){ System.out.println("B"); }
}
class C extends B {
    public static void main(String[] args){
        A obj = new C();
        obj.call();
    }
}

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

```
}
class B extends A {
    B(int a){ System.out.println("B"); }
    public static void main(String[] args){ new B(10); }
}
```

17. class A {

```
    A(){ System.out.println("A"); }
}
class B extends A {
    B(){ System.out.println("B"); }
}
class C extends B {
    C(){ System.out.println("C"); }
}
class Test {
    public static void main(String[] args){ new C(); }
}
```

18. class Demo {

```
    public static void main(String[] args){
        try{
            int a = 10/0;
        }
        catch(ArithmeticException e){
            System.out.println("AE");
        }
    }
}
```

19. class Test {

```
    public static void main(String[] args){
        try{
            try{ String s = null; s.length(); }
            catch(NullPointerException e)
{ System.out.println("NPE"); }
            int a = 5/0;
        }
        catch(Exception e){ System.out.println("EX"); }
    }
}
```

```
20. class A {
    static void m1(){ System.out.println("A"); }
}
class B extends A {
    static void m1(){ System.out.println("B"); }
    public static void main(String[] args){ A.m1(); B.m1(); }
}
```

```
21. class A extends Exception {}
class Test {
    static void m() throws A { throw new A(); }
    public static void main(String[] args){
        try{ m(); }
        catch(A e){ System.out.println("Caught"); }
    }
}
```

```
22. interface I1 { int a = 5; }
interface I2 { int a = 10; }
class Test {
    public static void main(String[] args){
        System.out.println(I1.a + " " + I2.a);
    }
}
```

```
23. class P {
    void show(){ System.out.println("P"); }
}
class Q extends P {
    void show(){ System.out.println("Q"); }
}
class Test {
    public static void main(String[] args){
        P p = new P();
        Q q = new Q();
        P r = new Q();
        p.show(); q.show(); r.show();
    }
}

24. class Demo {
    final int x;
    Demo(){ x = 20; }
    public static void main(String[] args){
        Demo d = new Demo();
        System.out.println(d.x);
    }
}

25. class Test {
    public static void main(String[] args){
        try{
            int arr[] = new int[2];
            System.out.println(arr[5]);
        }catch(ArrayIndexOutOfBoundsException e){
            System.out.println("AIOOB");
        }
    }
}
```

1.50
2.3
3.B
4.10 5
5.Q
6.A
7.int
8.Parent
Child
9.Hello
10.A
     B
     C
11.A
12.Block
Main
13.A
14.m1
15.B
16.A
B
17.A
B
C
18.AE
19.NPE
EX

20.A
B
21.Caught
22.5 10
23.P
Q
Q
24.20
25.AIOOB