Nombre: Juan Ivan Velazquez Cabello
Semana: Tarea semana 3

Java Cuestionario 1

1. Excepciones de Java permitida*

A. org.springframework.web.client.RestClientException
B. No conozco la respuesta
C. java.lang.NumberFormatException
D. com.bbva.apx.exception.db.NoResultException
E. com.bbva.elara.utility.interbackend.cics.exceptions.BusinessException

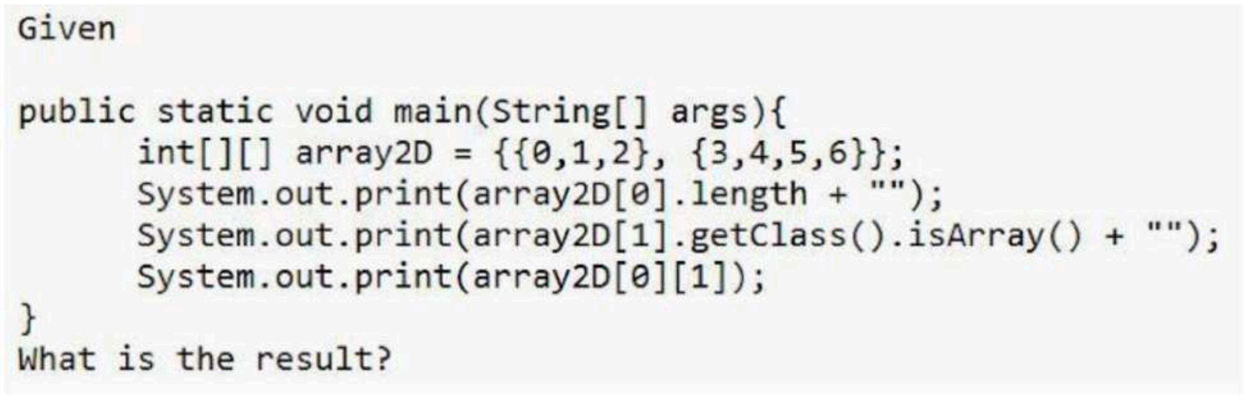
2. Which three are bad practices?

A. Checking for ArrayIndexOutOfBoundsException and ensuring that the program can recover if one occurs.
B. Checking for FileNotFoundException to inform a user that a filename entered is not valid.
C. Checking for Error and, if necessary, restarting the program to ensure that users are unaware of problems.
D. Checking for ArrayIndexOutOfBoundsException when iterating through an array to determine when all elements.-have been visited.

3. Indica a JUnit que la propiedad que usa esta anotación es una simulación y, por lo tanto, se inicializa como tal y es susceptible de ser inyectada por @InjectMocks.

A. Mockito
B. Mock
C. Inject
D. InjectMock

4.



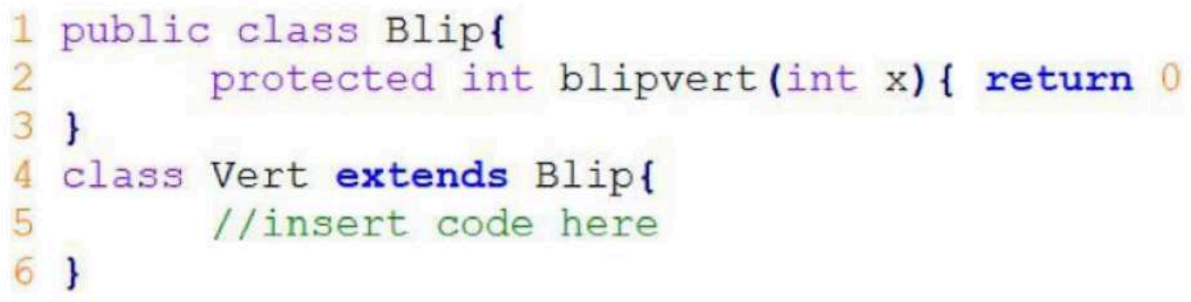
```
Given

public static void main(String[] args){
      int[][] array2D = {{0,1,2}, {3,4,5,6}};
      System.out.print(array2D[0].length + "");
      System.out.print(array2D[1].getClass().isArray() + "");
      System.out.print(array2D[0][1]);
}
What is the result?
```

A. 3false3
B. 3false1
C. 2false1
D. 3true1
E. 2true3

5. Which two statements are true?

A. An interface CANNOT be extended by another interface.
B. An abstract class can be extended by a concrete class.
C. An abstract class CANNOT be extended by an abstract class. An interface can be extended by an abstract class.
D. An abstract class can implement an interface
E. An abstract class can be extended by an interface.

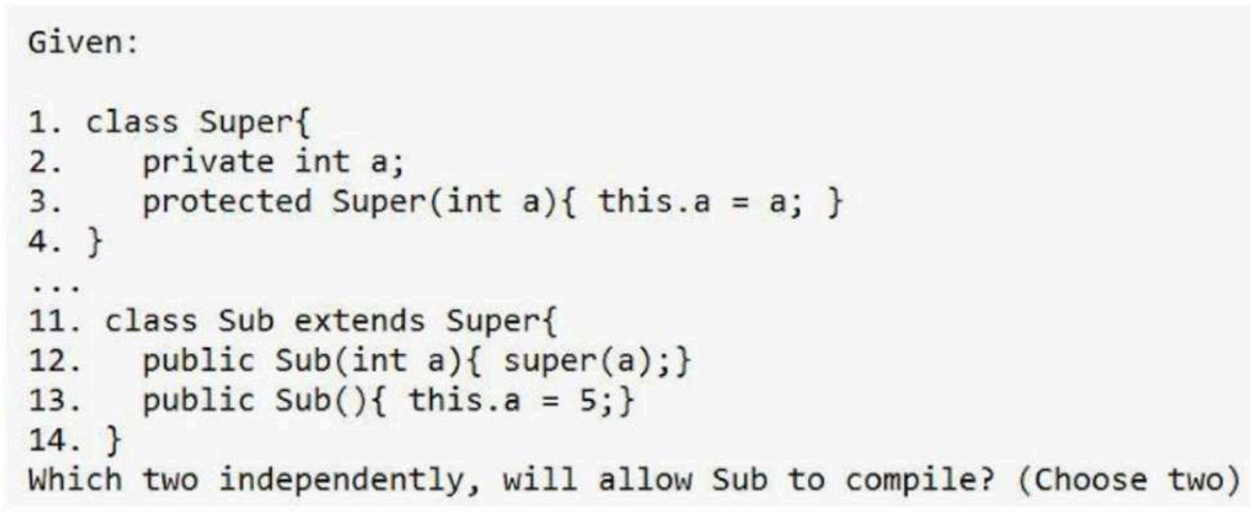
6. Which five methods, inserted independently at line 5, will compile? (Choose five)



```
1 public class Blip{
2         protected int blipvert(int x){ return 0
3 }
4 class Vert extends Blip{
5         //insert code here
6 }
```

A. private int blipvert(long x) { return 0; }
B. protected int blipvert(long x) { return 0; }
C. protected long blipvert(int x, int y) { return 0; }
D. public int blipvert(int x) { return 0; }
E. private int blipvert(int x) { return 0; }
F. protected long blipvert(int x) { return 0; }
G. protected long blipvert(long x) { return 0; }

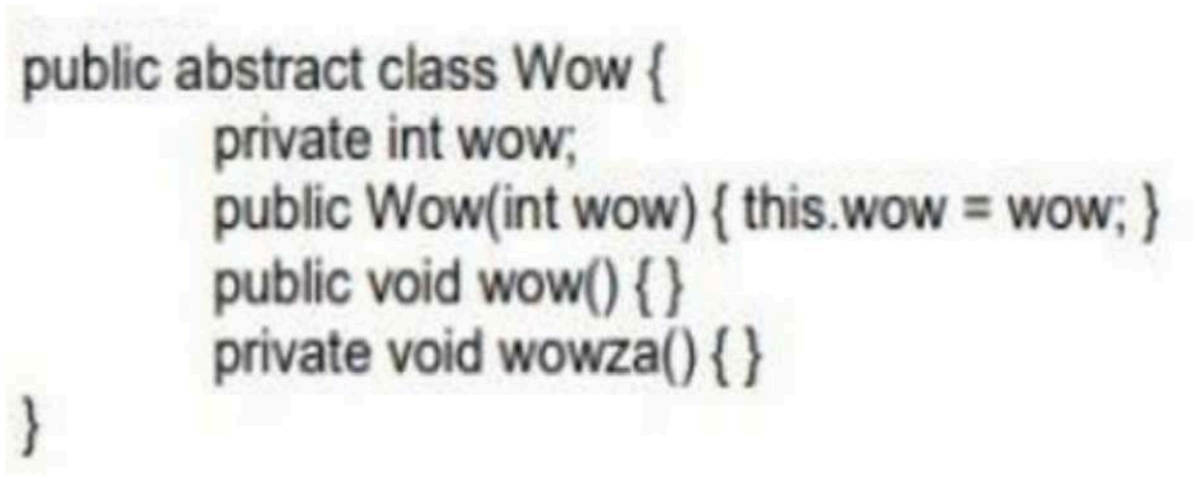
7.



```
Given:

1. class Super{
2.    private int a;
3.    protected Super(int a){ this.a = a; }
4. }
...
11. class Sub extends Super{
12.    public Sub(int a){ super(a);}
13.    public Sub(){ this.a = 5;}
14. }
Which two independently, will allow Sub to compile? (Choose two)
```

A. Change line 2 to: public int a;
B. Change line 13 to: public Sub(){ super(5);}
C. Change line 2 to: protected int a;
D. Change line 13 to: public Sub(){ this(5);}
E. Change line 13 to: public Sub(){super(a);}

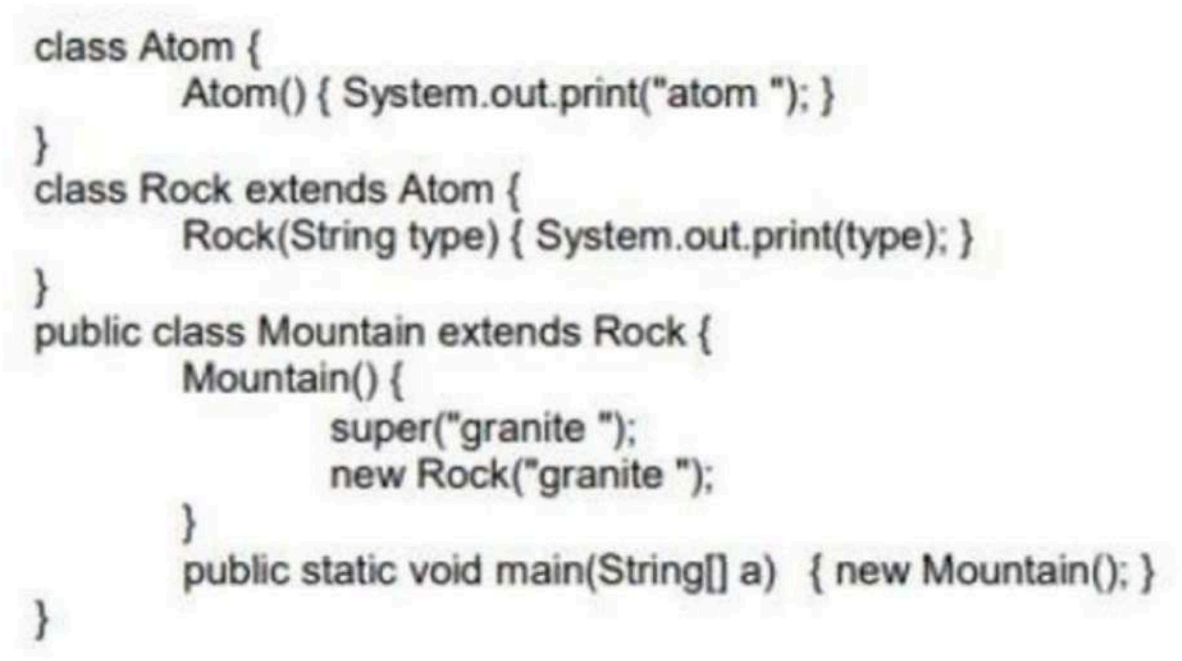
8. What is true about the class Wow?



```
public abstract class Wow {
        private int wow;
        public Wow(int wow) { this.wow = wow; }
        public void wow() { }
        private void wowza() { }
}
```

A. It compiles without error.
B. It does not compile because an abstract class cannot have private methods
C. It does not compile because an abstract class cannot have instance variables.
D. It does not compile because an abstract class must have at least one abstract method.
E. It does not compile because an abstract class must have a constructor with no arguments.

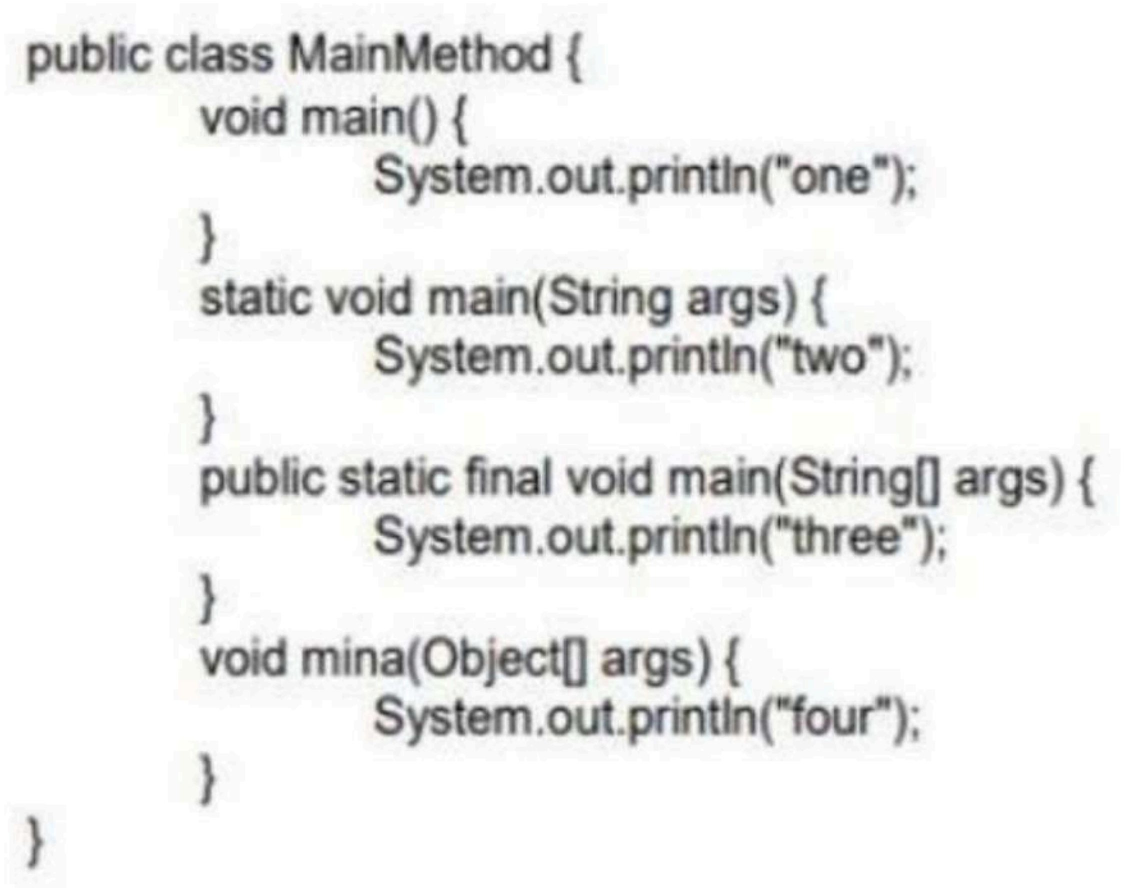
10- What is the result?



```
class Atom {
        Atom() { System.out.print("atom "); }
}
class Rock extends Atom {
        Rock(String type) { System.out.print(type); }
}
public class Mountain extends Rock {
        Mountain() {
                super("granite ");
                new Rock("granite ");
        }
        public static void main(String[] a)  { new Mountain(); }
}
```

A. Compilation fails.
B. Atom granite.
C. Granite granite.
D. Atom granite granite.
E. An exception isthrown at runtime
F. Atom granite atom granite

11 What is printed out when the program is excuted?

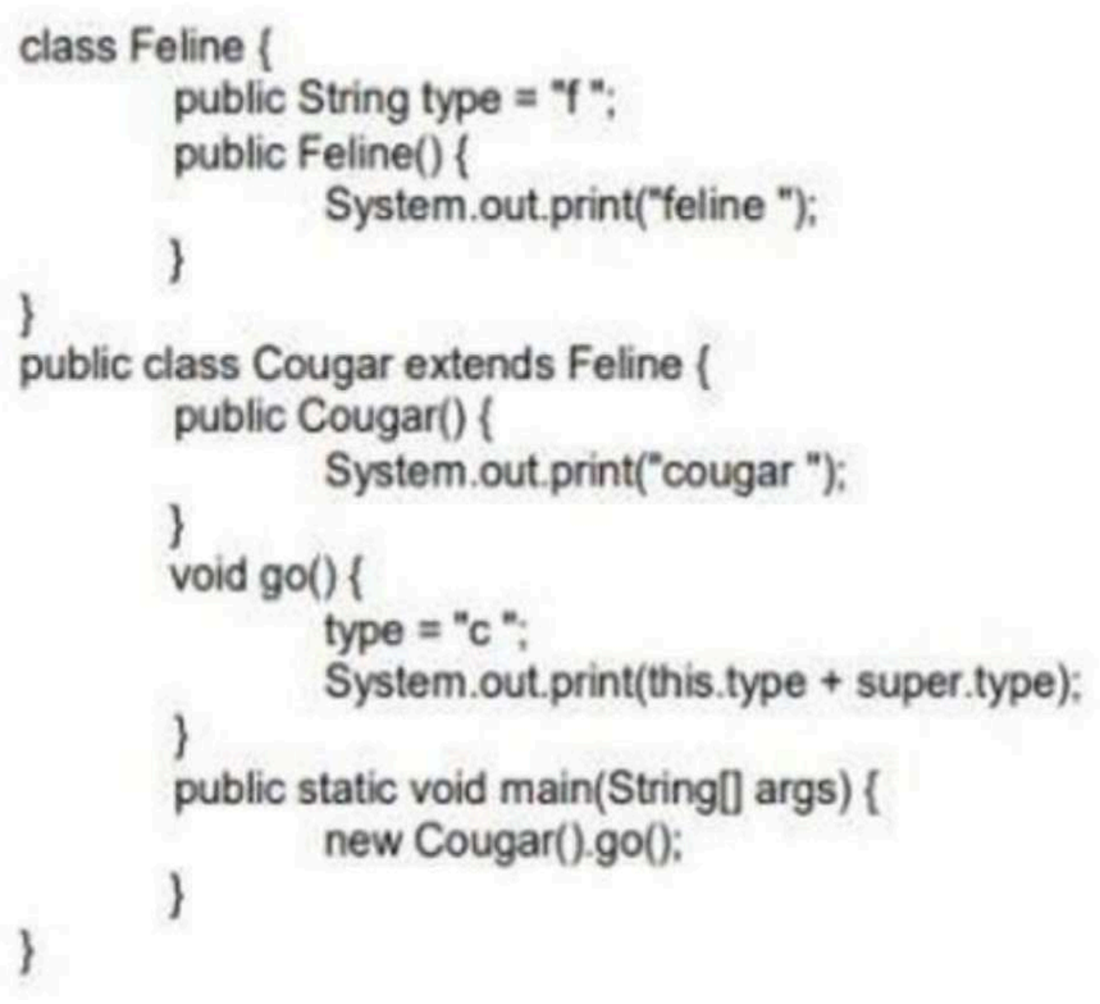


```
public class MainMethod {
        void main() {
                System.out.println("one");
        }
        static void main(String args) {
                System.out.println("two");
        }
        public static final void main(String[] args) {
                System.out.println("three");
        }
        void mina(Object[] args) {
                System.out.println("four");
        }
}
```

A. One

B. Two
C. Three
D. Four
E. There is no output

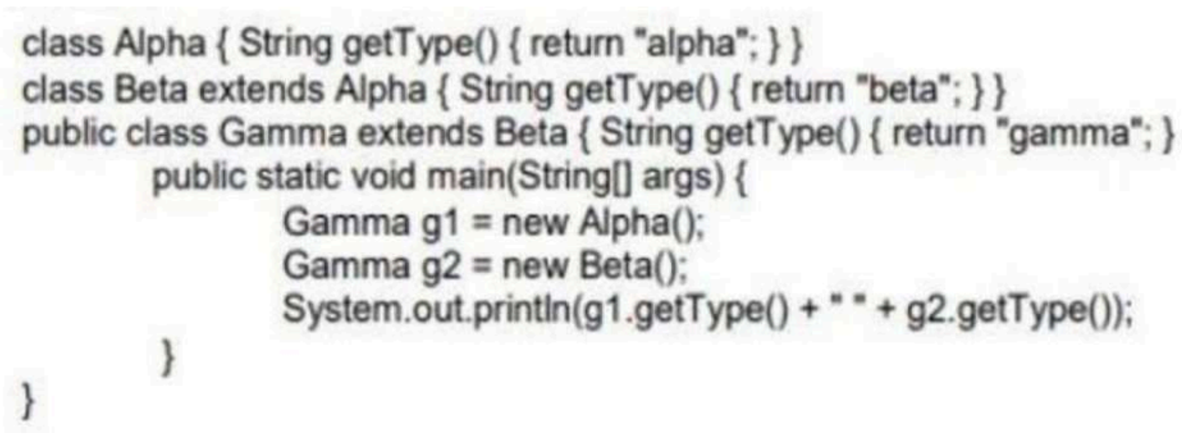
12 What is the result?



```
class Feline {
        public String type = "f ";
        public Feline() {
                System.out.print("feline ");
        }
}
public class Cougar extends Feline {
        public Cougar() {
                System.out.print("cougar ");
        }
        void go() {
                type = "c ";
                System.out.print(this.type + super.type);
        }
        public static void main(String[] args) {
                new Cougar().go();
        }
}
```

A. Cougar c f.
B. Feline cougar c f.
C. Feline cougar c c.
D. Compilation fails.

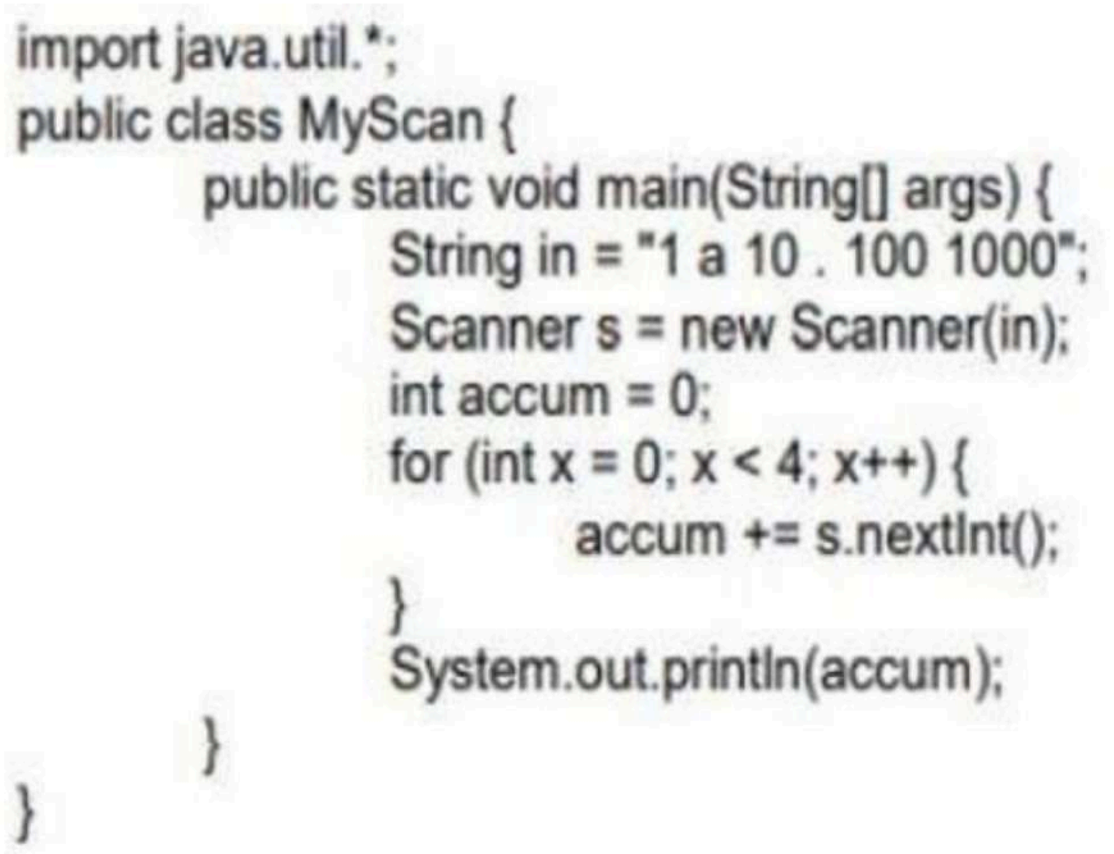
12. What is the result?



```
class Alpha { String getType() { return "alpha"; } }
class Beta extends Alpha { String getType() { return "beta"; } }
public class Gamma extends Beta { String getType() { return "gamma"; }
          public static void main(String[] args) {
                    Gamma g1 = new Alpha();
                    Gamma g2 = new Beta();
                    System.out.println(g1.getType() + " " + g2.getType());
          }
}
```

A. Alpha beta
B. Beta beta.
C. Gamma gamma.
D. Compilation fails.

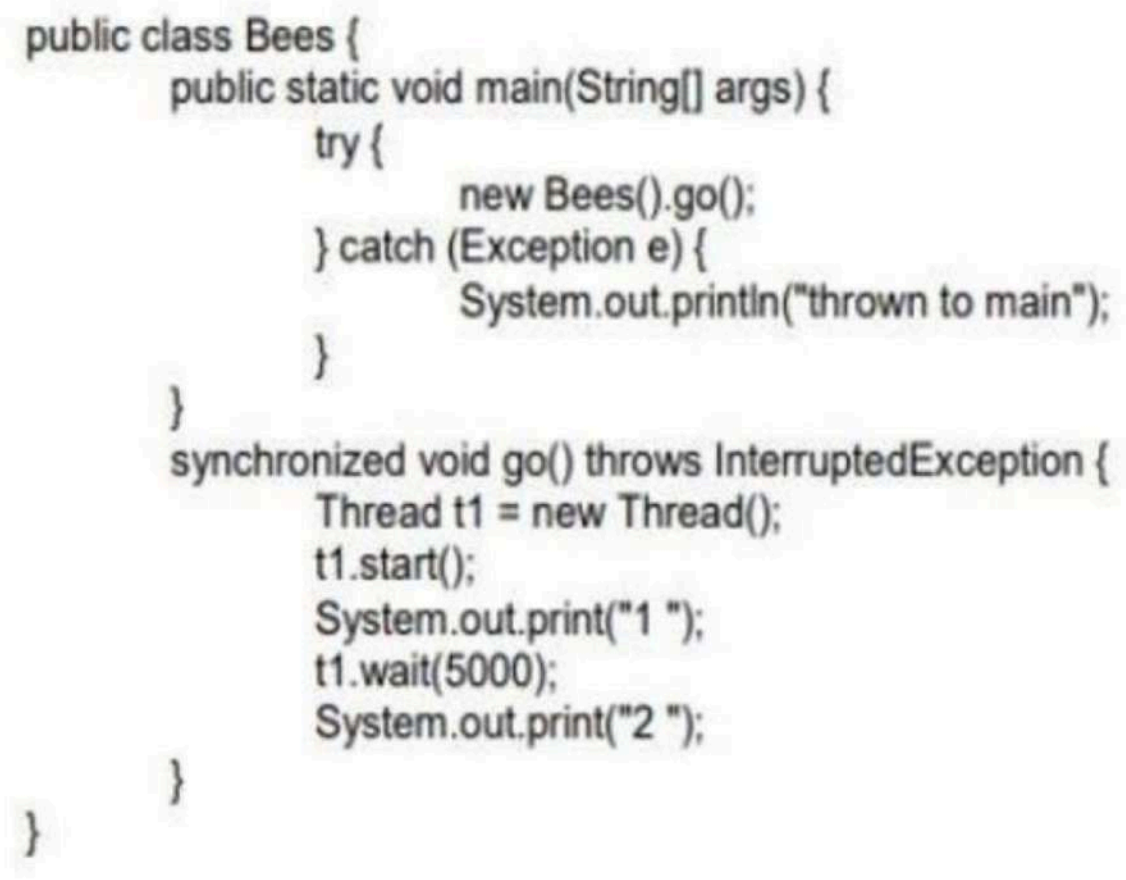
13. What is the result?



```
import java.util.*;
public class MyScan {
          public static void main(String[] args) {
                    String in = "1 a 10 . 100 1000";
                    Scanner s = new Scanner(in);
                    int accum = 0;
                    for (int x = 0; x < 4; x++) {
                              accum += s.nextInt();
                    }
                    System.out.println(accum);
          }
}
```

A. 11
B. 11
C. 1111
D. An exception is thrown at runtime

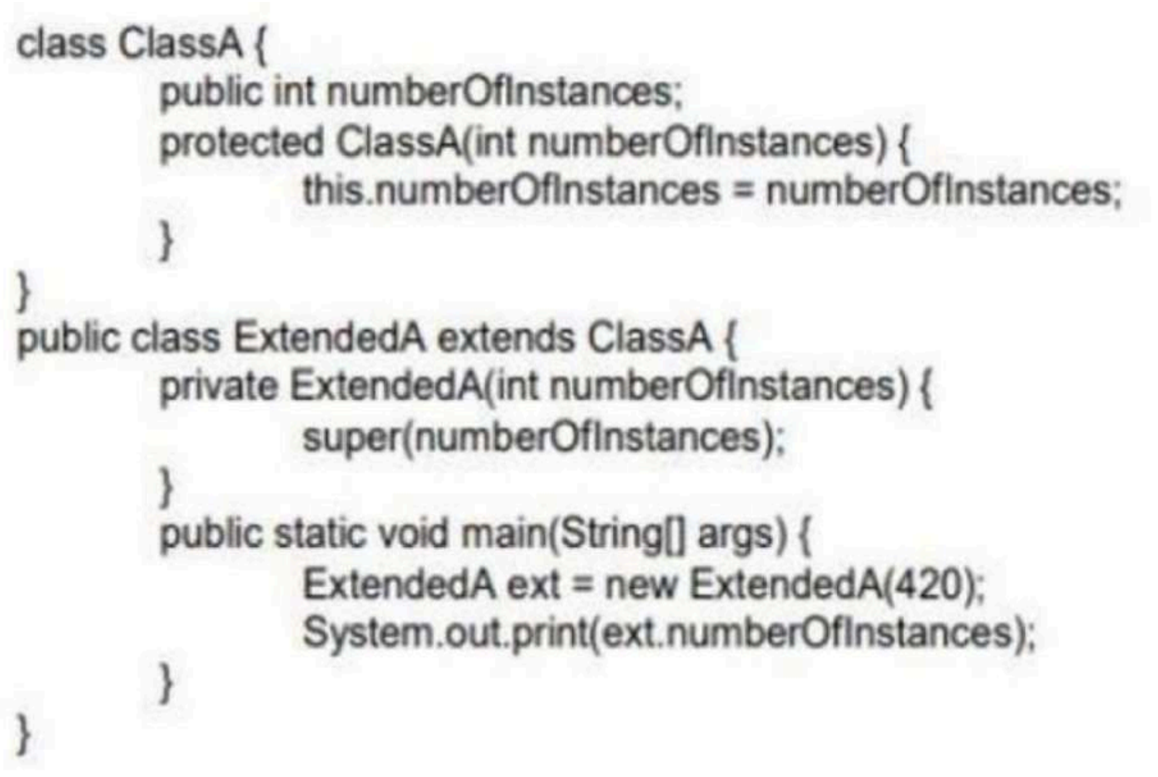
14. What is the result?



```
public class Bees {
        public static void main(String[] args) {
                try {
                        new Bees().go();
                } catch (Exception e) {
                        System.out.println("thrown to main");
                }
        }
        synchronized void go() throws InterruptedException {
                Thread t1 = new Thread();
                t1.start();
                System.out.print("1 ");
                t1.wait(5000);
                System.out.print("2 ");
        }
}
```

A. The program prints 1 then 2 after 5 seconds.
B. The program prints: 1 thrown to main.
C. The program prints: 1 2 thrown to main.
D. The program prints:1 then t1 waits for its notification.

15. Which statement is true?



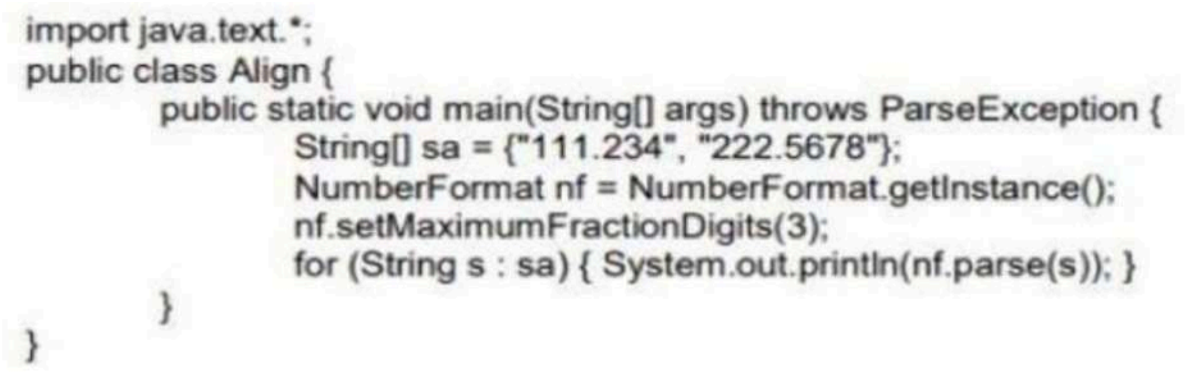
```
class ClassA {
        public int numberOfInstances;
        protected ClassA(int numberOfInstances) {
                this.numberOfInstances = numberOfInstances;
        }
}
public class ExtendedA extends ClassA {
        private ExtendedA(int numberOfInstances) {
                super(numberOfInstances);
        }
        public static void main(String[] args) {
                ExtendedA ext = new ExtendedA(420);
                System.out.print(ext.numberOfInstances);
        }
}
```

A. 420 is the output.
B. An exception isthrown at runtime.
C. All constructors must be declared public.
D. Constructors CANNOT use the private modifier.
E. Constructors CANNOT use the protected modifier

16. The SINGLETON pattern allows:

A. Have a single instance of a class and this instance cannot be used by other classes
B. Having a single instance of a class, while allowing all classes to have access to that instance.
C. Having a single instance of a class that can only be accessed by the first method that calls it.

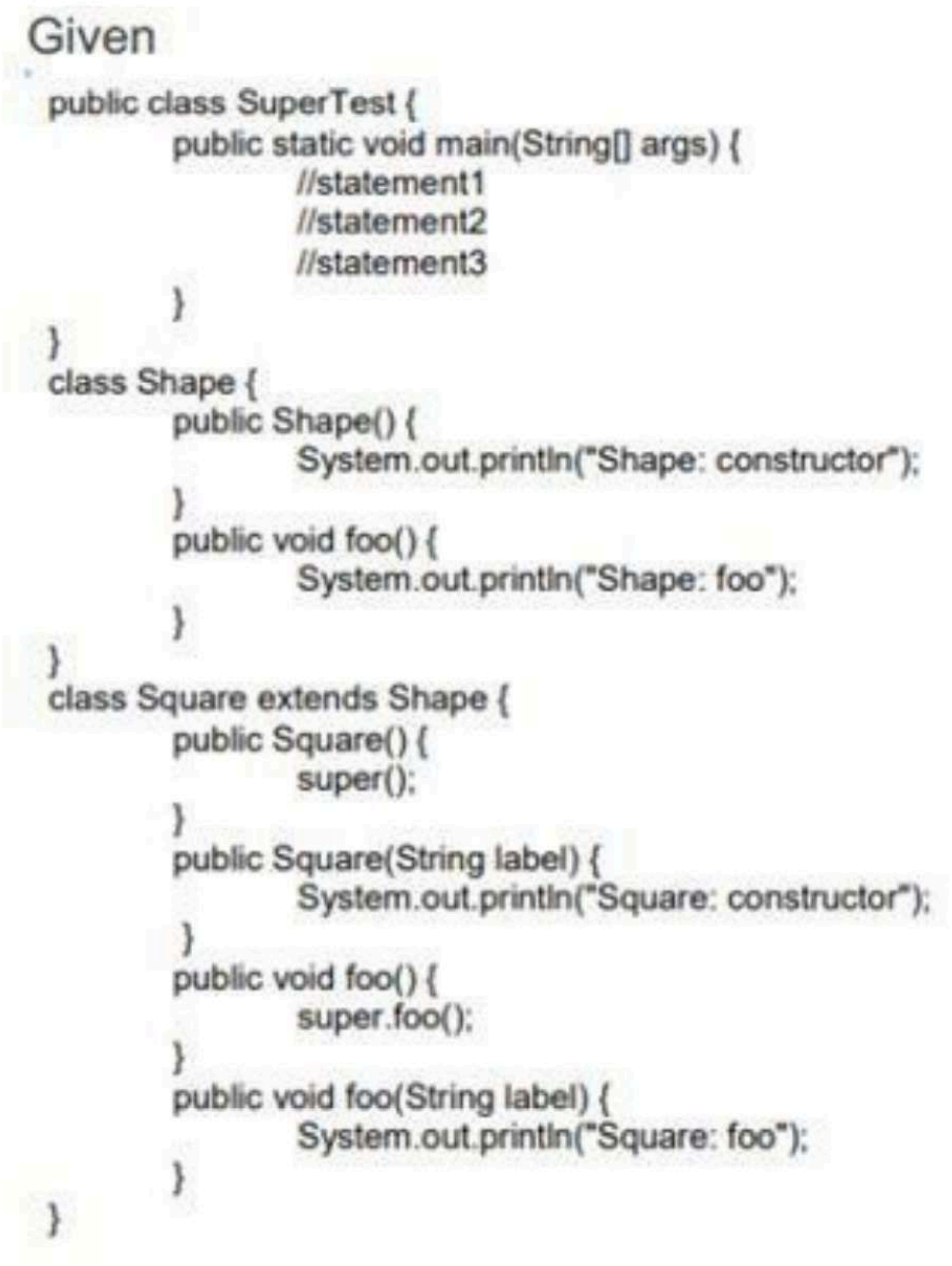
17. What is the result?



```
import java.text.*;
public class Align {
        public static void main(String[] args) throws ParseException {
                String[] sa = {"111.234", "222.5678"};
                NumberFormat nf = NumberFormat.getInstance();
                nf.setMaximumFractionDigits(3);
                for (String s : sa) { System.out.println(nf.parse(s)); }
        }
}
```

A. 111.234 222.567
B. 111.234 222.568
C. 111.234 222.5678
D. An exception isthrown at runtime.

18. What is the result? What should statement1, statement2 and statement 3 be respectively, in order to produce the result. : shape: constructor, shape: foo, square: foo

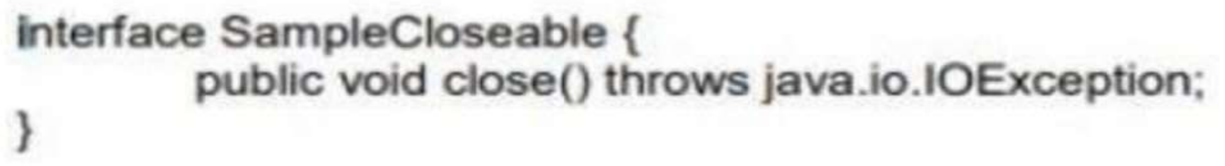


Given

```
public class SuperTest {
        public static void main(String[] args) {
                //statement1
                //statement2
                //statement3
        }
}
class Shape {
        public Shape() {
                System.out.println("Shape: constructor");
        }
        public void foo() {
                System.out.println("Shape: foo");
        }
}
class Square extends Shape {
        public Square() {
                super();
        }
        public Square(String label) {
                System.out.println("Square: constructor");
         }
        public void foo() {
                super.foo();
        }
        public void foo(String label) {
                System.out.println("Square: foo");
        }
}
```

A. Square square = new Square ("bar"); square.foo ("bar"); square.foo();
B. Square square = new Square ("bar"); square.foo ("bar"); square.foo ("bar");
C. Square square = new Square (); square.foo ();square.foo(bar);
D. Square square = new Square (); square.foo (); square.foo("bar");
E. Square square = new Square (); square.foo (); square.foo ();

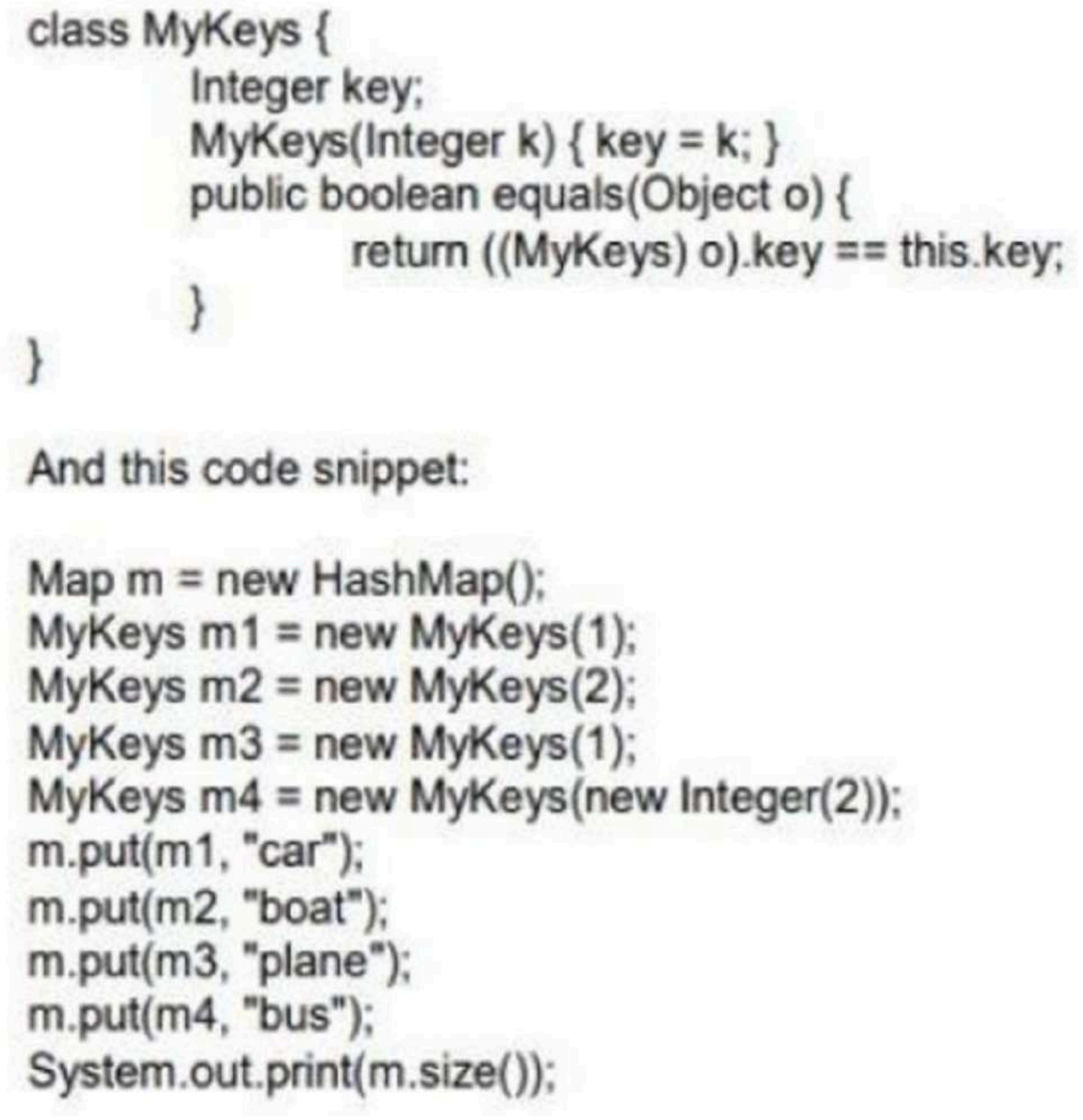
19. Which three implementations are valid?



```
interface SampleCloseable {
        public void close() throws java.io.IOException;
}
```

A. class Test implements SampleCloseable { public void close() throws java.io.IOException { // do something } }
B. class Test Implements SampleCloseable { public void close() throws Exception { // do something } }
C. class Test implements SampleCloseable { public void close() throws FileNotFoundException {// do something } }
D. class Test extends SampleCloseable { public void close() throws java.io.IOException { // do something } }
E. class Test implements SampleCloseable { public void close() { // do something }}

20. What is the result?



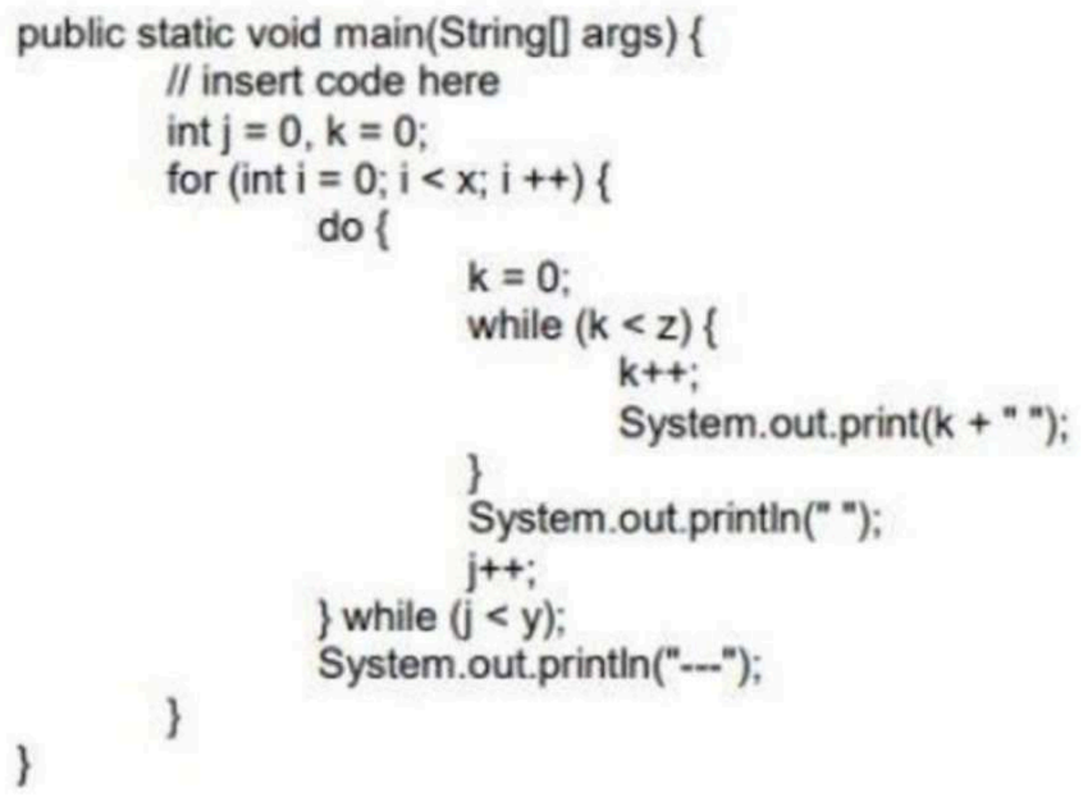
```
class MyKeys {
        Integer key;
        MyKeys(Integer k) { key = k; }
        public boolean equals(Object o) {
                return ((MyKeys) o).key == this.key;
        }
}
```

And this code snippet:

```
Map m = new HashMap();
MyKeys m1 = new MyKeys(1);
MyKeys m2 = new MyKeys(2);
MyKeys m3 = new MyKeys(1);
MyKeys m4 = new MyKeys(new Integer(2));
m.put(m1, "car");
m.put(m2, "boat");
m.put(m3, "plane");
m.put(m4, "bus");
System.out.print(m.size());
```

A. 2
B. 3
[✓ C. 4]
D. Compilation fails.

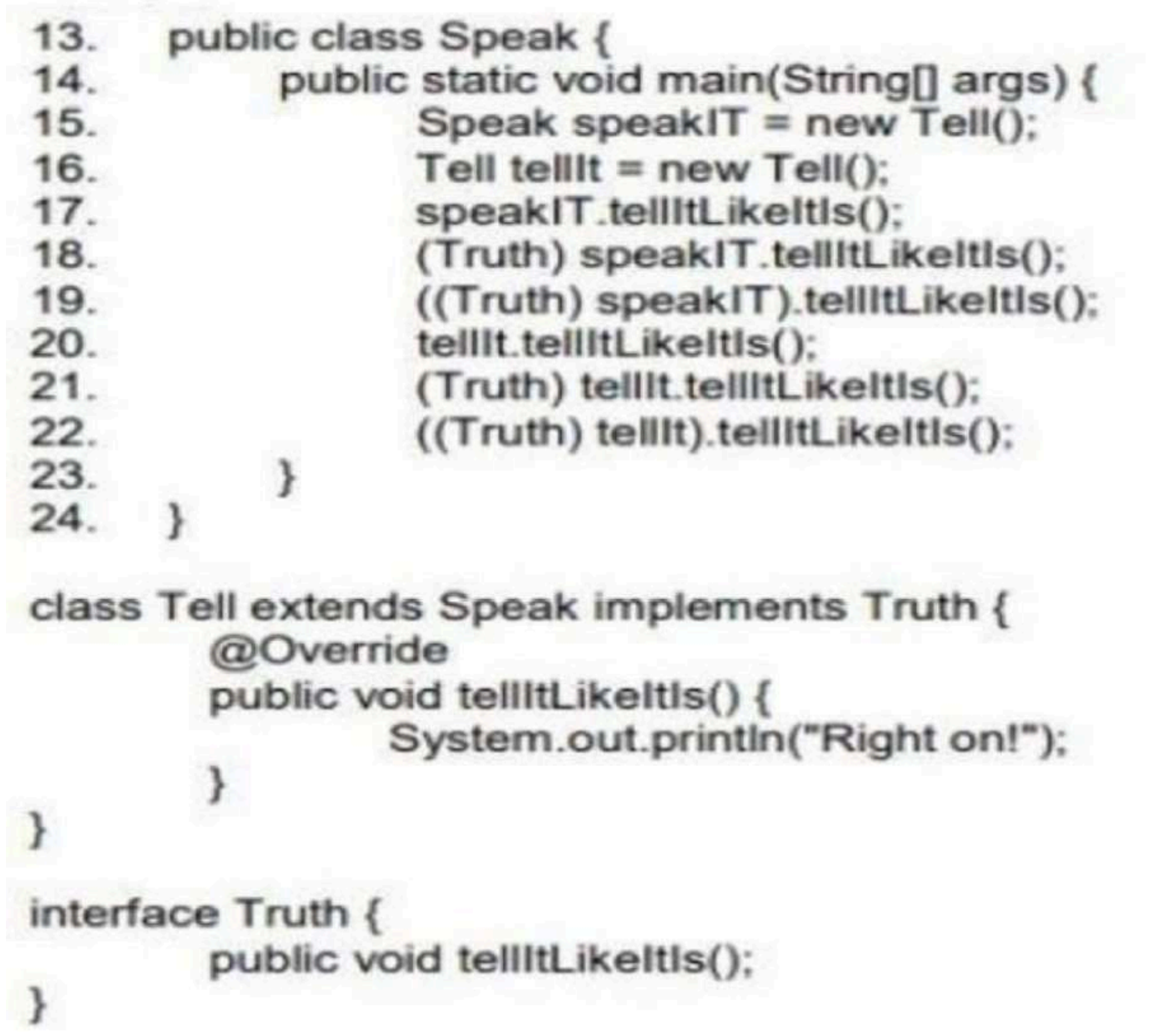
21. What value of x, y, z will produce the following result? 1234,1234,1234 -----, 1234, ------



```
public static void main(String[] args) {
        // insert code here
        int j = 0, k = 0;
        for (int i = 0; i < x; i ++) {
                do {
                        k = 0;
                        while (k < z) {
                                k++;
                                System.out.print(k + " ");
                        }
                        System.out.println(" ");
                        j++;
                } while (j < y);
                System.out.println("---");
        }
}
```

A. int x = 4, y = 3, z = 2;
B. int x = 3, y = 2, z = 3;
C. int x = 2, y = 3, z = 3
D. int x = 2, y = 3, z = 4;
E. int x = 4, y = 2, z= 3;

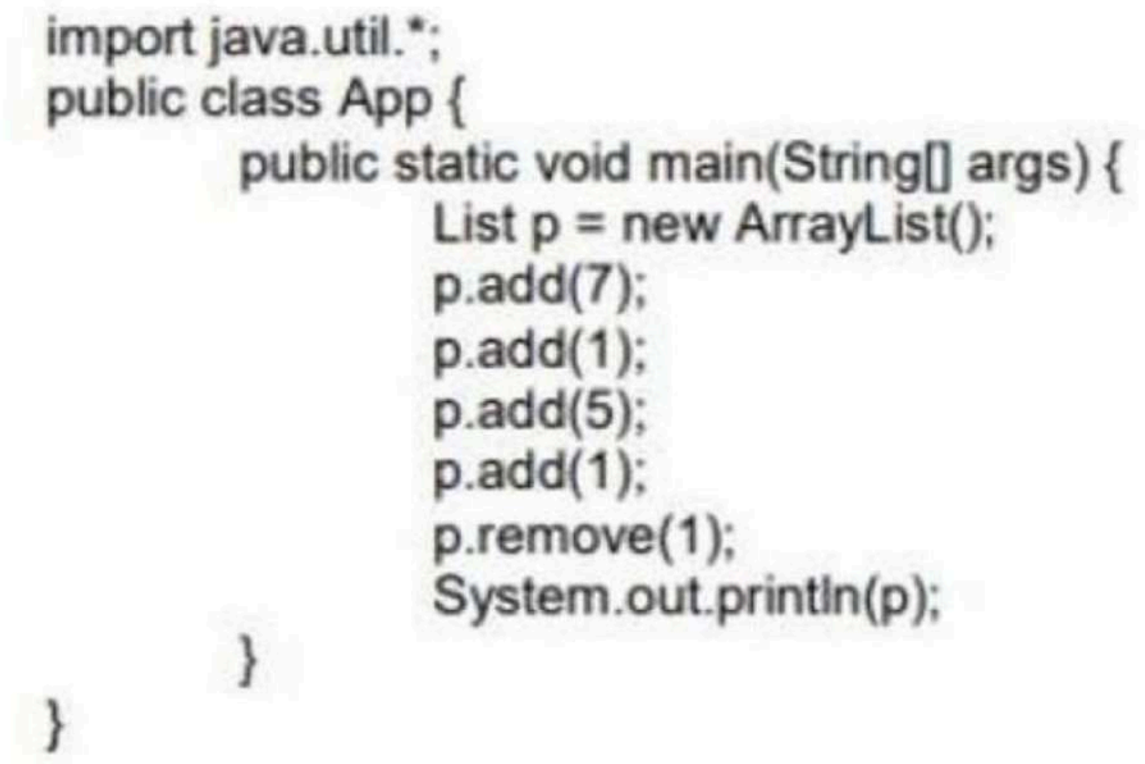
22. Which three lines will compile and output "Right



```
13.    public class Speak {
14.        public static void main(String[] args) {
15.            Speak speakIT = new Tell();
16.            Tell tellIt = new Tell();
17.            speakIT.tellItLikeItIs();
18.            (Truth) speakIT.tellItLikeItIs();
19.            ((Truth) speakIT).tellItLikeItIs();
20.            tellIt.tellItLikeItIs();
21.            (Truth) tellIt.tellItLikeItIs();
22.            ((Truth) tellIt).tellItLikeItIs();
23.        }
24.    }

class Tell extends Speak implements Truth {
        @Override
        public void tellItLikeItIs() {
                System.out.println("Right on!");
        }
}

interface Truth {
        public void tellItLikeItIs();
}
```

A. Line 17
B. Line 18
C. Line 19
D. Line 20
E. Line 21
F. Line 22

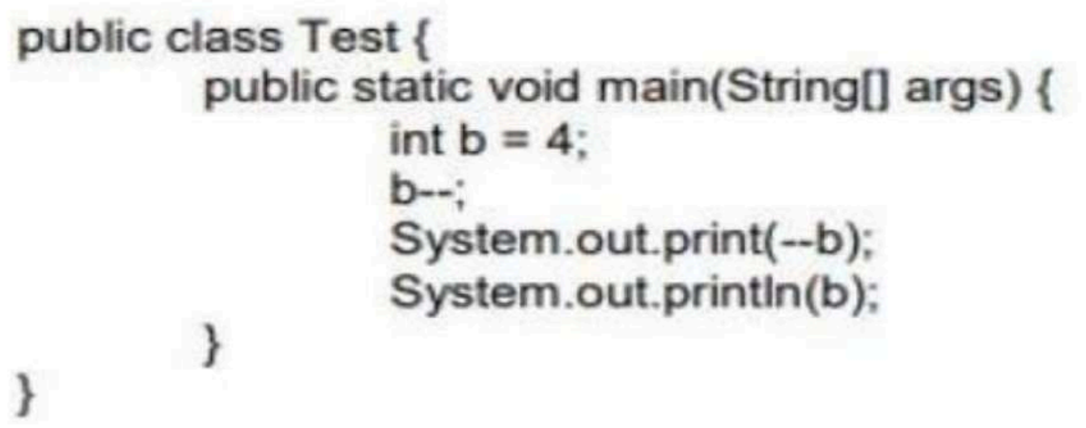
23. ¿Cuál es el resultado?



```
import java.util.*;
public class App {
        public static void main(String[] args) {
                List p = new ArrayList();
                p.add(7);
                p.add(1);
                p.add(5);
                p.add(1);
                p.remove(1);
                System.out.println(p);
        }
}
```

A. [7, 5]
B. [7, 1]
C. [7, 5, 1]
D. [7, 1, 5, 1]

24. What is the result?



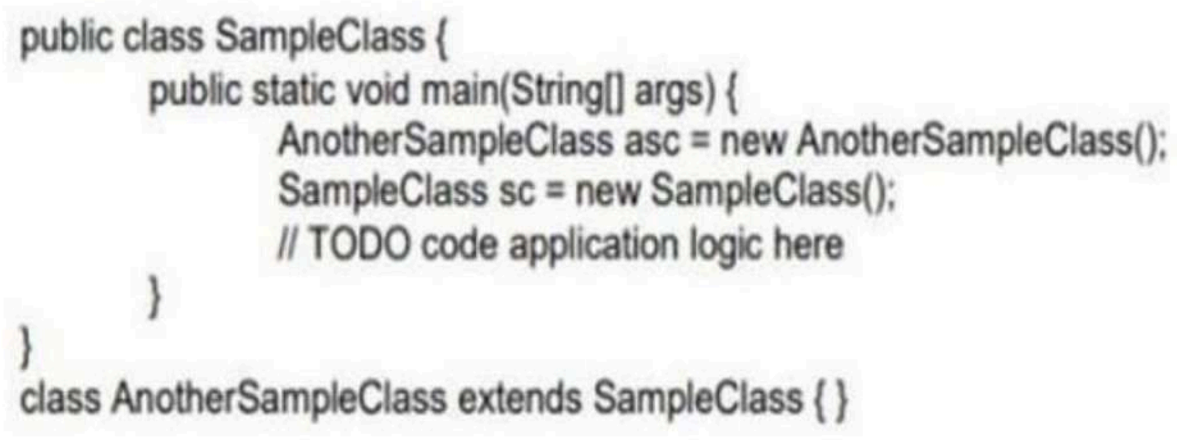
```
public class Test {
        public static void main(String[] args) {
                int b = 4;
                b--;
                System.out.print(--b);
                System.out.println(b);
        }
}
```

A. 22
B. 12
C. 32
D. 33

25. In Java the difference between throws and throw Is:

A. Throws throws an exception and throw indicates the type of exception that the method.
B. Throws Is used in methods and throw in constructors.
C. Throws indicate the type of exception that the method does not handle and throw an exception.

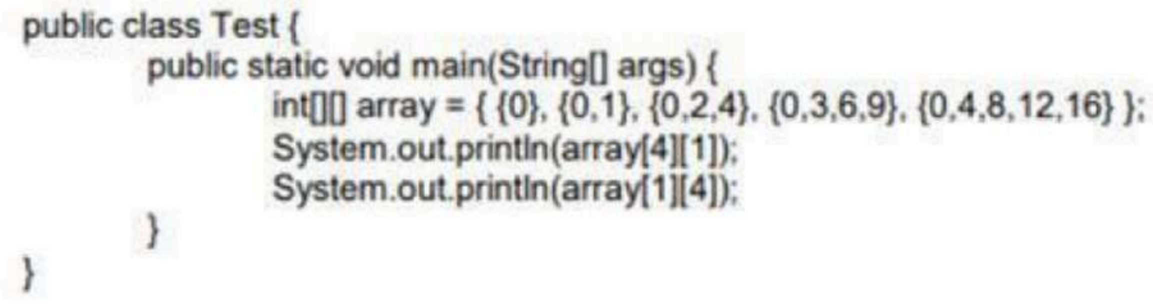
26. Which statement, when inserted into line " // TODO code application logic here", is valid in compilation time change?



```
public class SampleClass {
        public static void main(String[] args) {
                AnotherSampleClass asc = new AnotherSampleClass();
                SampleClass sc = new SampleClass();
                // TODO code application logic here
        }
}
class AnotherSampleClass extends SampleClass { }
```

A. asc = sc;
B. sc = asc;
C. asc = (Object)sc;
D. asc= sc.clone();

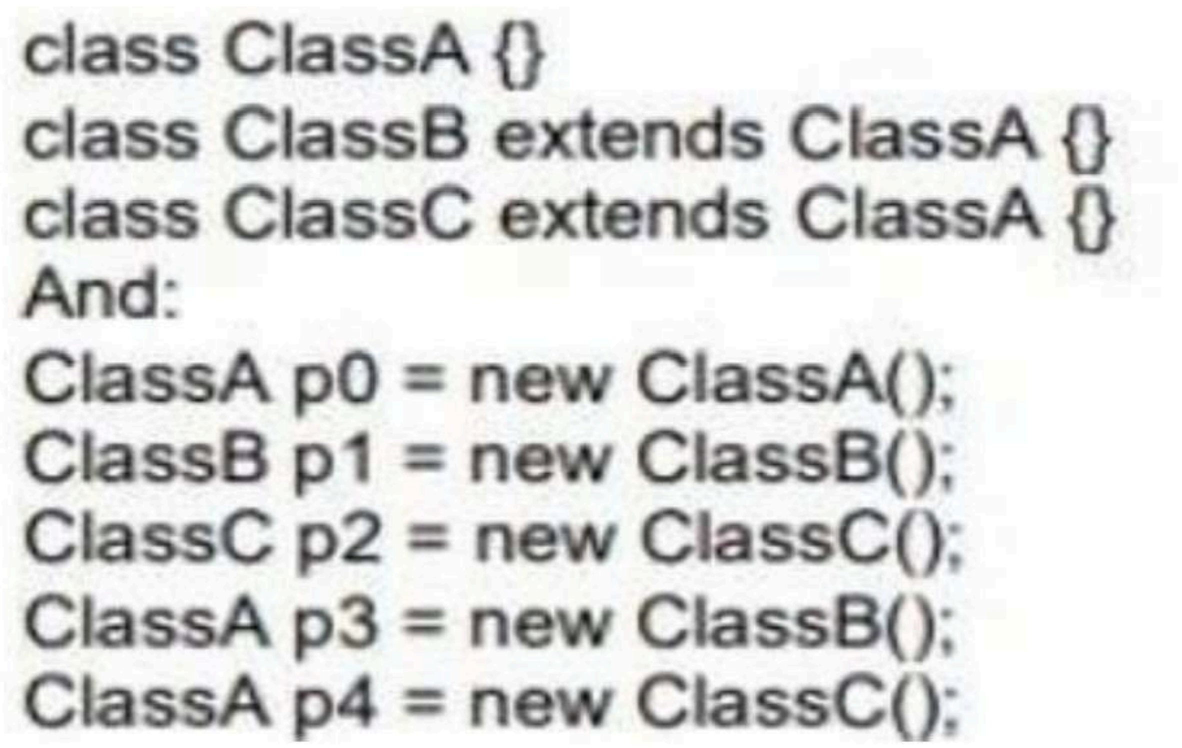
27. What is the result?



```
public class Test {
        public static void main(String[] args) {
                int[][] array = { {0}, {0,1}, {0,2,4}, {0,3,6,9}, {0,4,8,12,16} };
                System.out.println(array[4][1]);
                System.out.println(array[1][4]);
        }
}
```

A. 4 Null.
B. Null 4.
C. An IllegalArgumentExceptionis thrown at run time.
D. 4 An ArrayIndexOutOfBoundsException is thrown at run time.

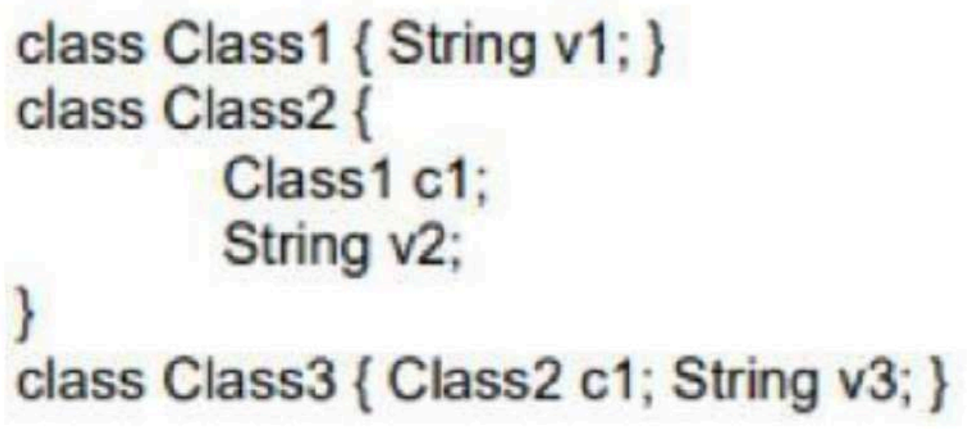
28. Which three are valid? (Choose three)



```
class ClassA {}
class ClassB extends ClassA {}
class ClassC extends ClassA {}
And:
ClassA p0 = new ClassA();
ClassB p1 = new ClassB();
ClassC p2 = new ClassC();
ClassA p3 = new ClassB();
ClassA p4 = new ClassC();
```

A. **p0 = p1;**
B. p1 = p2;
C. p2 = p4;
D. p2 = (ClassC)p1;
E. **p1 = (ClassB)p3;**
F. **p2 = (ClassC)p4;**

29. Which three options correctly describe the relationship between the classes?

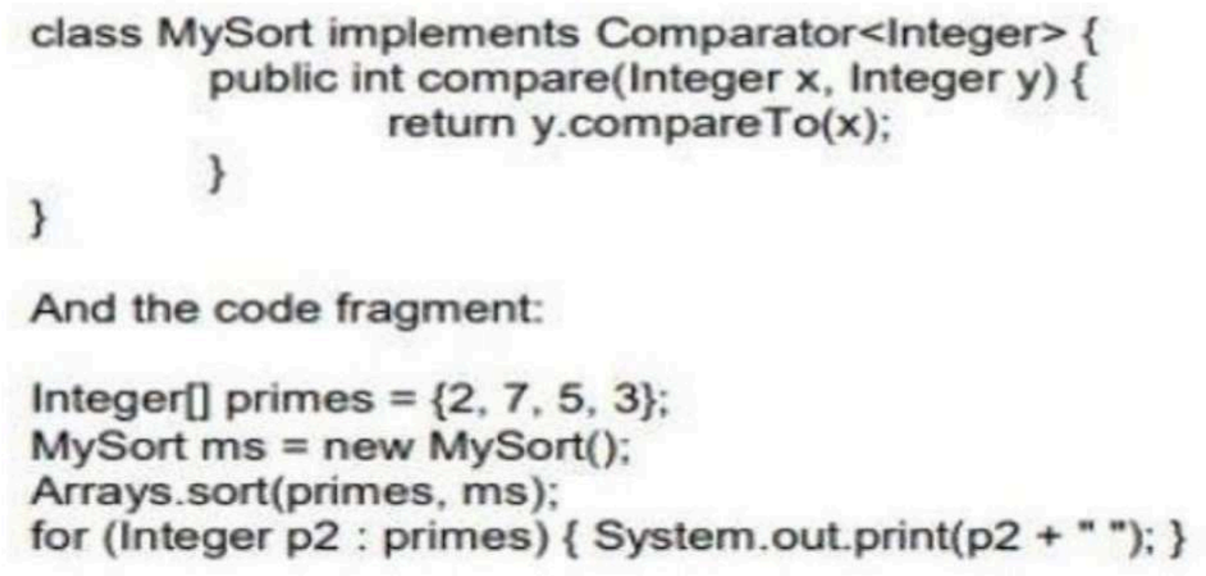


```
class Class1 { String v1; }
class Class2 {
            Class1 c1;
            String v2;
}
class Class3 { Class2 c1; String v3; }
```

A. Class2 has - a v3.
B. Class1 has - a v2.

C. **Class2 has - a v2.**
D. **Class3 has - a v1.**
E. Class2 has - a Class3
F. **Class2 has - aClass1**

30. What is the result?



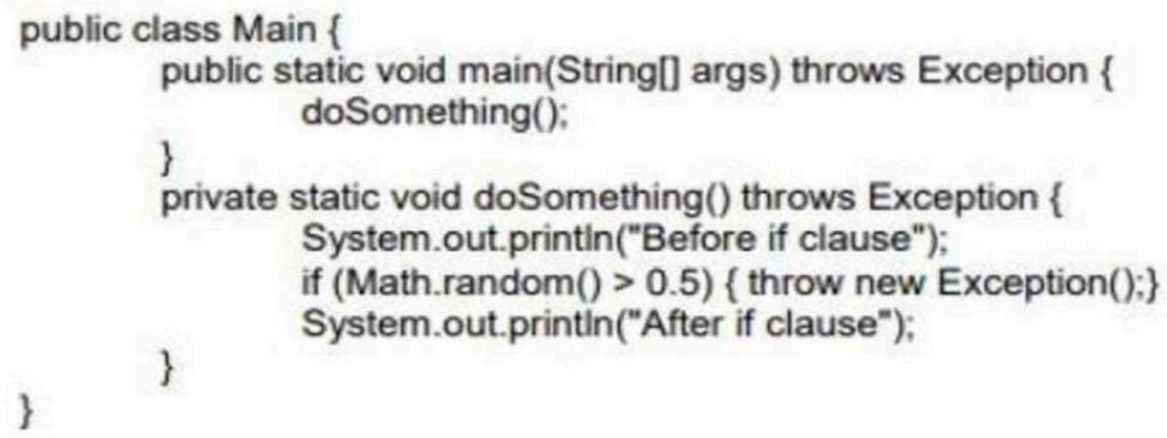
```
class MySort implements Comparator<Integer> {
        public int compare(Integer x, Integer y) {
                return y.compareTo(x);
        }
}
```

And the code fragment:

```
Integer[] primes = {2, 7, 5, 3};
MySort ms = new MySort();
Arrays.sort(primes, ms);
for (Integer p2 : primes) { System.out.print(p2 + " "); }
```

A. 2 3 5 7
B. 2 7 5 3
C. **7 5 3 2**
D. Compilation fails.

31. Which two possible outputs?

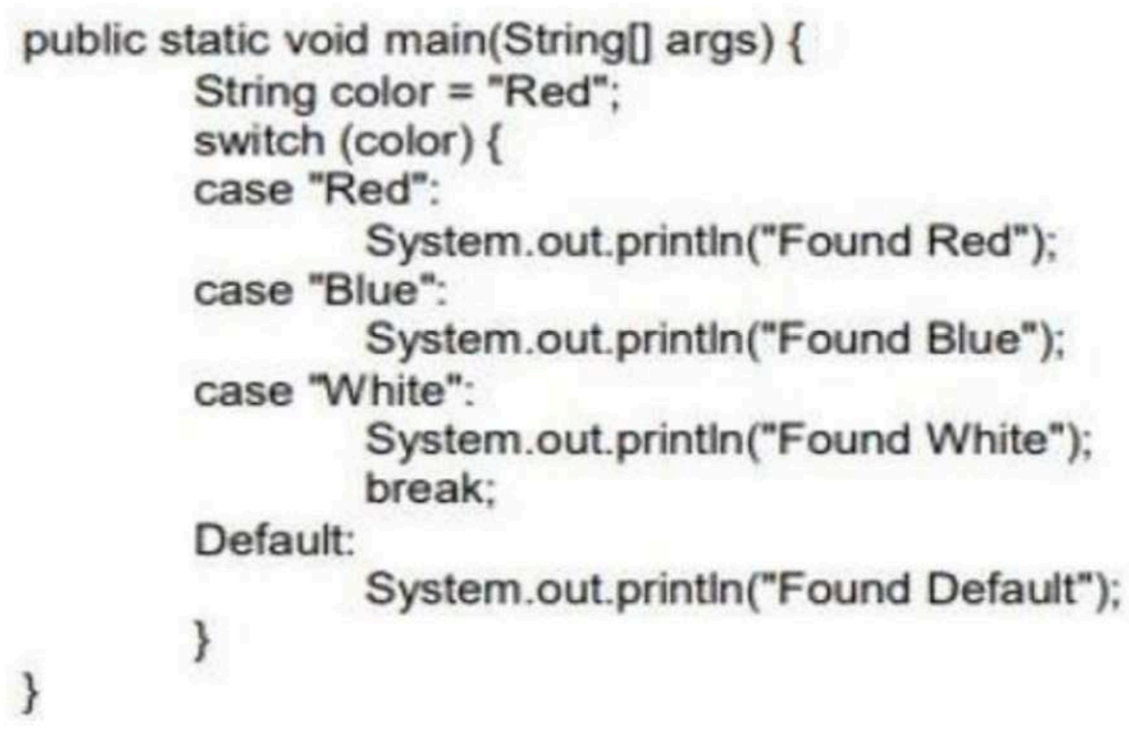


```
public class Main {
        public static void main(String[] args) throws Exception {
                doSomething();
        }
        private static void doSomething() throws Exception {
                System.out.println("Before if clause");
                if (Math.random() > 0.5) { throw new Exception();}
                System.out.println("After if clause");
        }
}
```

A. Before if clause Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15).
B. Before if clause Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15) After if clause.

C. **Exception in thread "main" java.lang.Exception at Main.doSomething (Main.java:21) at Main.main (Main.java:15).**
D. **Before if clause After if clause.**

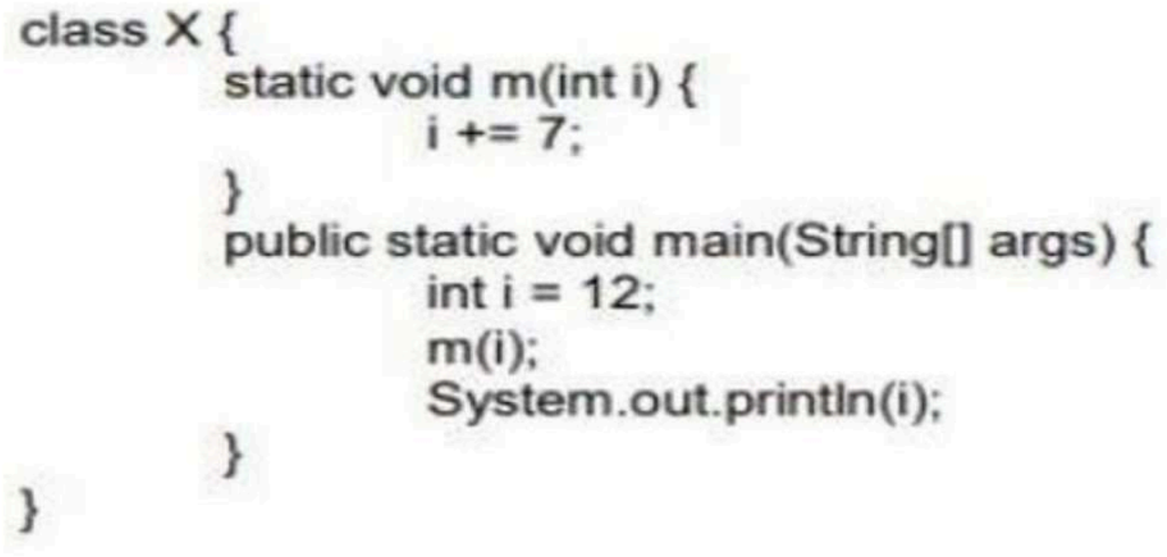
32. What is the result?



```
public static void main(String[] args) {
            String color = "Red";
            switch (color) {
            case "Red":
                        System.out.println("Found Red");
            case "Blue":
                        System.out.println("Found Blue");
            case "White":
                        System.out.println("Found White");
                        break;
            Default:
                        System.out.println("Found Default");
            }
}
```

A. Found Red.
B. Found Red Found Blue.
C. **Found Red Found Blue Found White.**
D. Found Red Found Blue Found White Found Default.

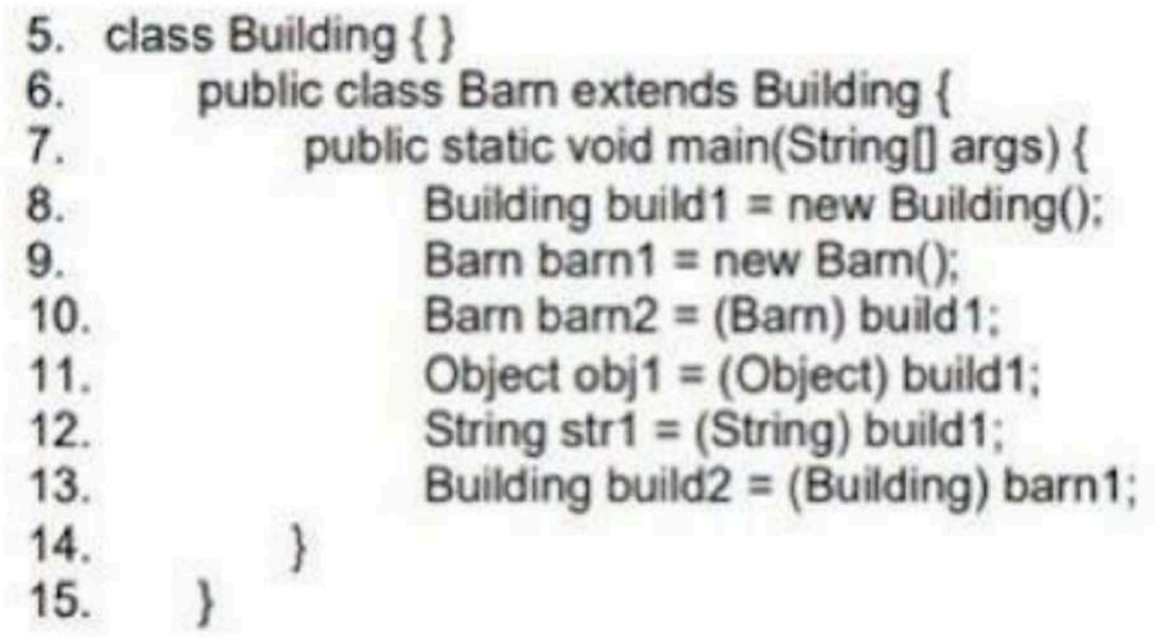
33. What is the result?



```
class X {
        static void m(int i) {
                i += 7;
        }
        public static void main(String[] args) {
                int i = 12;
                m(i);
                System.out.println(i);
        }
}
```

A. 7
B. 12
C. 19
D. Compilation fails.
E. An exception isthrown at run time

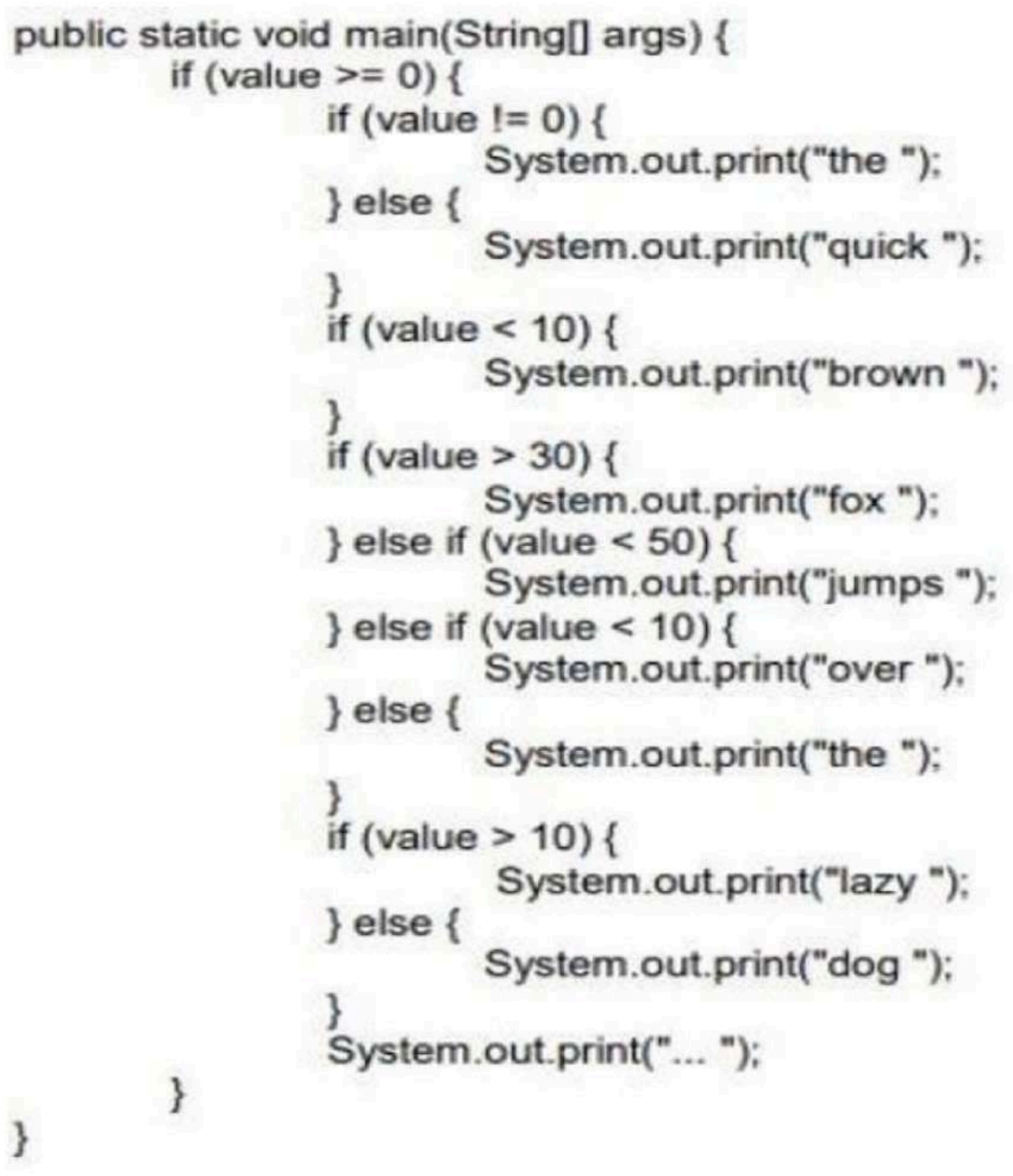
34. Which is true?



```
5.  class Building { }
6.      public class Barn extends Building {
7.          public static void main(String[] args) {
8.              Building build1 = new Building();
9.              Barn barn1 = new Barn();
10.             Barn barn2 = (Barn) build1;
11.             Object obj1 = (Object) build1;
12.             String str1 = (String) build1;
13.             Building build2 = (Building) barn1;
14.         }
15.     }
```

A. If line 10 is removed, the compilation succeeds.
B. If line 11 is removed, the compilation succeeds.
C. If line 12 is removed, the compilation succeeds.
D. If line 13 is removed, the compilation succeeds.
E. More than one line must be removed for compilation to succeed.

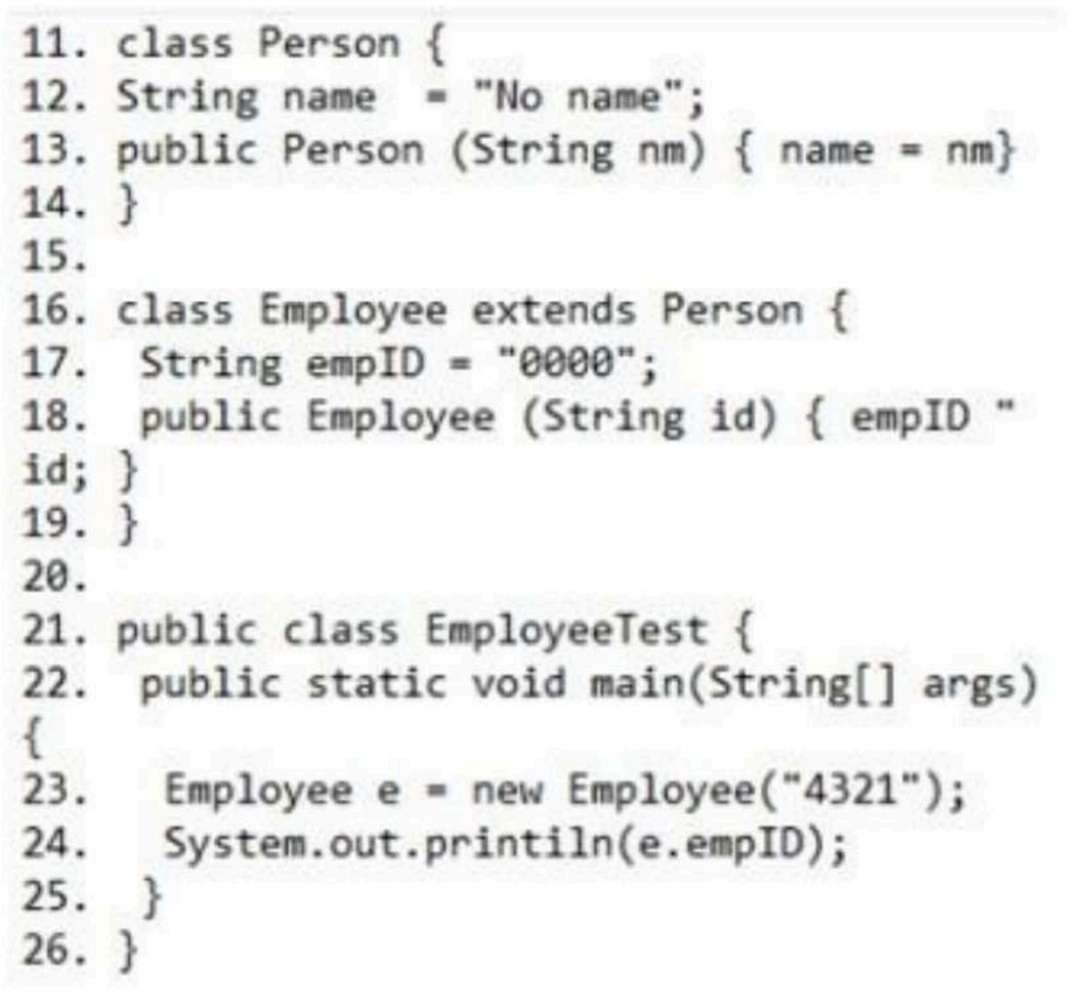
35. What is the result if the integer value is 33?



```
public static void main(String[] args) {
        if (value >= 0) {
                if (value != 0) {
                        System.out.print("the ");
                } else {
                        System.out.print("quick ");
                }
                if (value < 10) {
                        System.out.print("brown ");
                }
                if (value > 30) {
                        System.out.print("fox ");
                } else if (value < 50) {
                        System.out.print("jumps ");
                } else if (value < 10) {
                        System.out.print("over ");
                } else {
                        System.out.print("the ");
                }
                if (value > 10) {
                        System.out.print("lazy ");
                } else {
                        System.out.print("dog ");
                }
                System.out.print("... ");
        }
}
```

A. The fox jump lazy...
B. The fox lazy...
C. Quick fox over lazy…
D. Quick fox the…

36. What is the result?



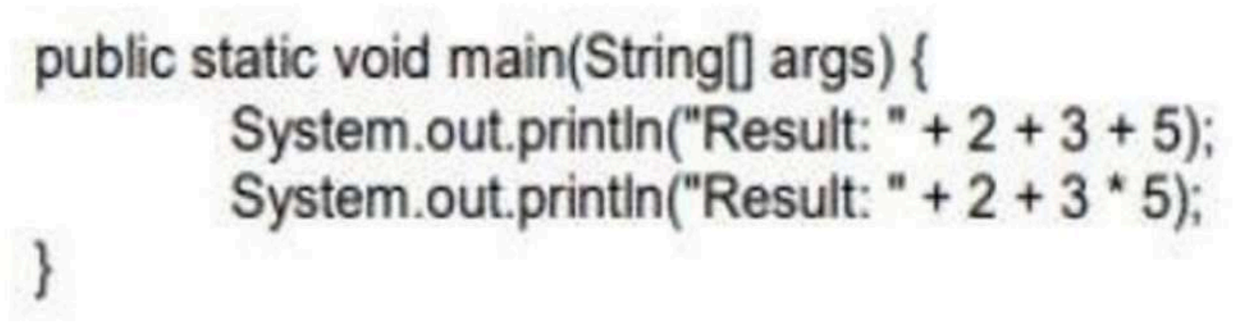
```
11. class Person {
12. String name  = "No name";
13. public Person (String nm) { name = nm}
14. }
15.
16. class Employee extends Person {
17.  String empID = "0000";
18.  public Employee (String id) { empID "
id; }
19. }
20.
21. public class EmployeeTest {
22.  public static void main(String[] args)
{
23.   Employee e = new Employee("4321");
24.   System.out.println(e.empID);
25.  }
26. }
```

A. 4321.
B. 0000.
C. An exception isthrown at runtime.
D. Compilation fails because of an error in line 18.

37. Which code fragment is illegal?

A. Class Base1 { abstract class Abs1{ } }
B. Abstract class Abs2 { void doit() { } }
C. class Base2{ abstract class Abs3extends Base2 { } }
D. class Base3 { abstract int var1 = 89; }

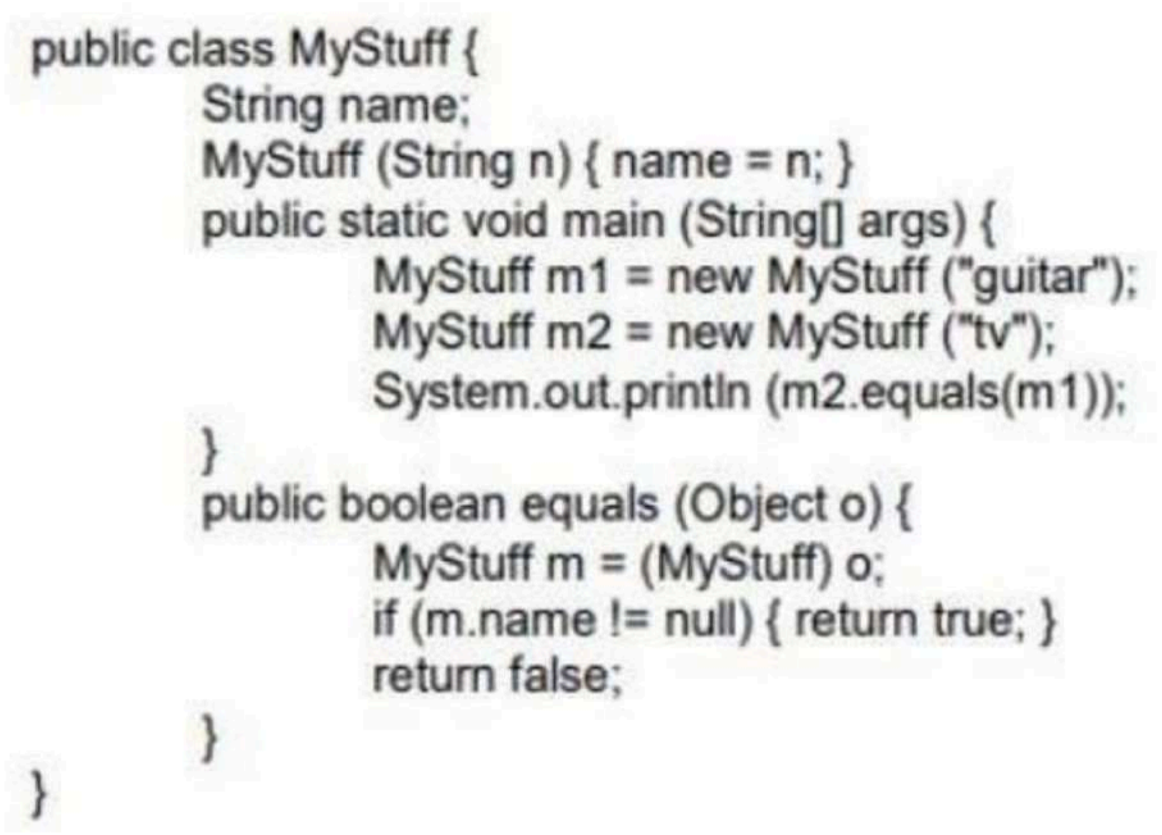
38. What is the result?



```
public static void main(String[] args) {
            System.out.println("Result: " + 2 + 3 + 5);
            System.out.println("Result: " + 2 + 3 * 5);
}
```

A. Result: 10 Result: 30
B. Result: 25 Result: 10
C. Result: 235 Result: 215
D. Result: 215 Result: 215
E. Compilation fails.

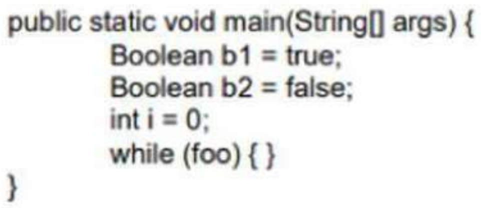
39. What is the result?



```
public class MyStuff {
        String name;
        MyStuff (String n) { name = n; }
        public static void main (String[] args) {
                MyStuff m1 = new MyStuff ("guitar");
                MyStuff m2 = new MyStuff ("tv");
                System.out.println (m2.equals(m1));
        }
        public boolean equals (Object o) {
                MyStuff m = (MyStuff) o;
                if (m.name != null) { return true; }
                return false;
        }
}
```

A. The output istrue and MyStufffulfills the Object.equals() contract
B. The output is false andMyStufffulfills the Object.equals() contract
C. The output is true and MyStuff does NOT fulfill the Object.equals() contract.
D. The output is false andMyStuff doesNOT fulfill the Object.equals() contract

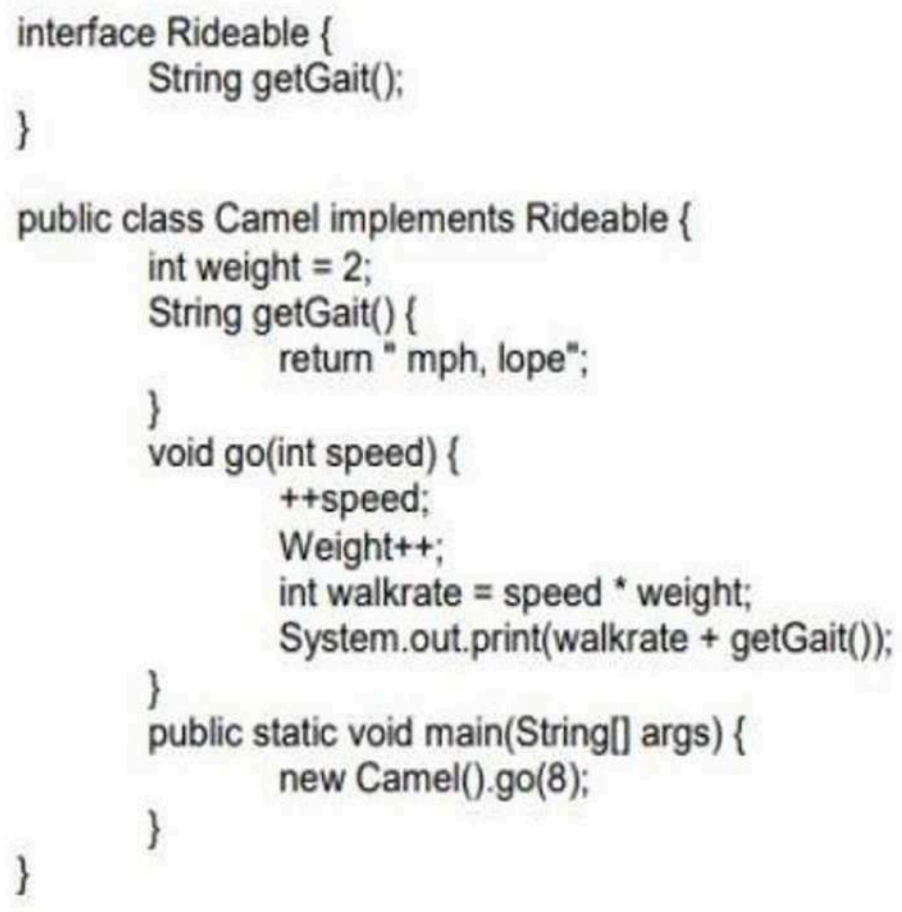
40. Which one is valid as a replacement for foo?



```
public static void main(String[] args) {
        Boolean b1 = true;
        Boolean b2 = false;
        int i = 0;
        while (foo) { }
}
```

A. b1.compareTo(b2)
B. i = 1
C. i == 2? -1:0
D. foo.equals("bar")

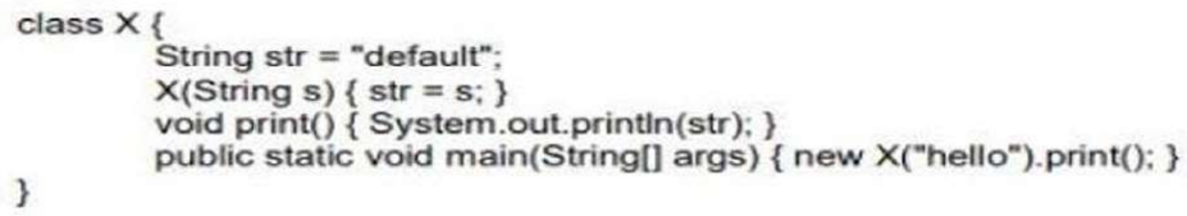
41. What is the result?



```
interface Rideable {
          String getGait();
}

public class Camel implements Rideable {
          int weight = 2;
          String getGait() {
                    return " mph, lope";
          }
          void go(int speed) {
                    ++speed;
                    Weight++;
                    int walkrate = speed * weight;
                    System.out.print(walkrate + getGait());
          }
          public static void main(String[] args) {
                    new Camel().go(8);
          }
}
```

A. 16 mph, lope
B. 24 mph, lope.
C. Compilation fails
D. 27 mph, lope.

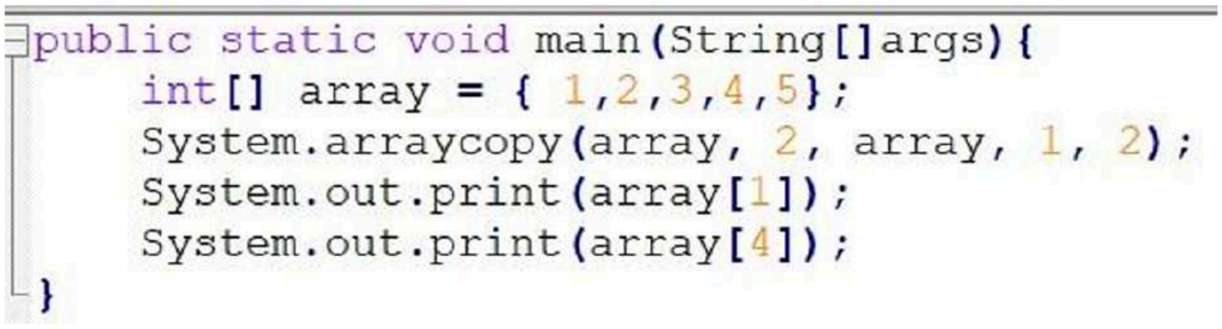
42. What is the result?



```
class X {
        String str = "default";
        X(String s) { str = s; }
        void print() { System.out.println(str); }
        public static void main(String[] args) { new X("hello").print(); }
}
```

A. **Hello**
B. Default
C. Compilation fails.
D. The program prints nothing.
E. An exception is thrown at run time.

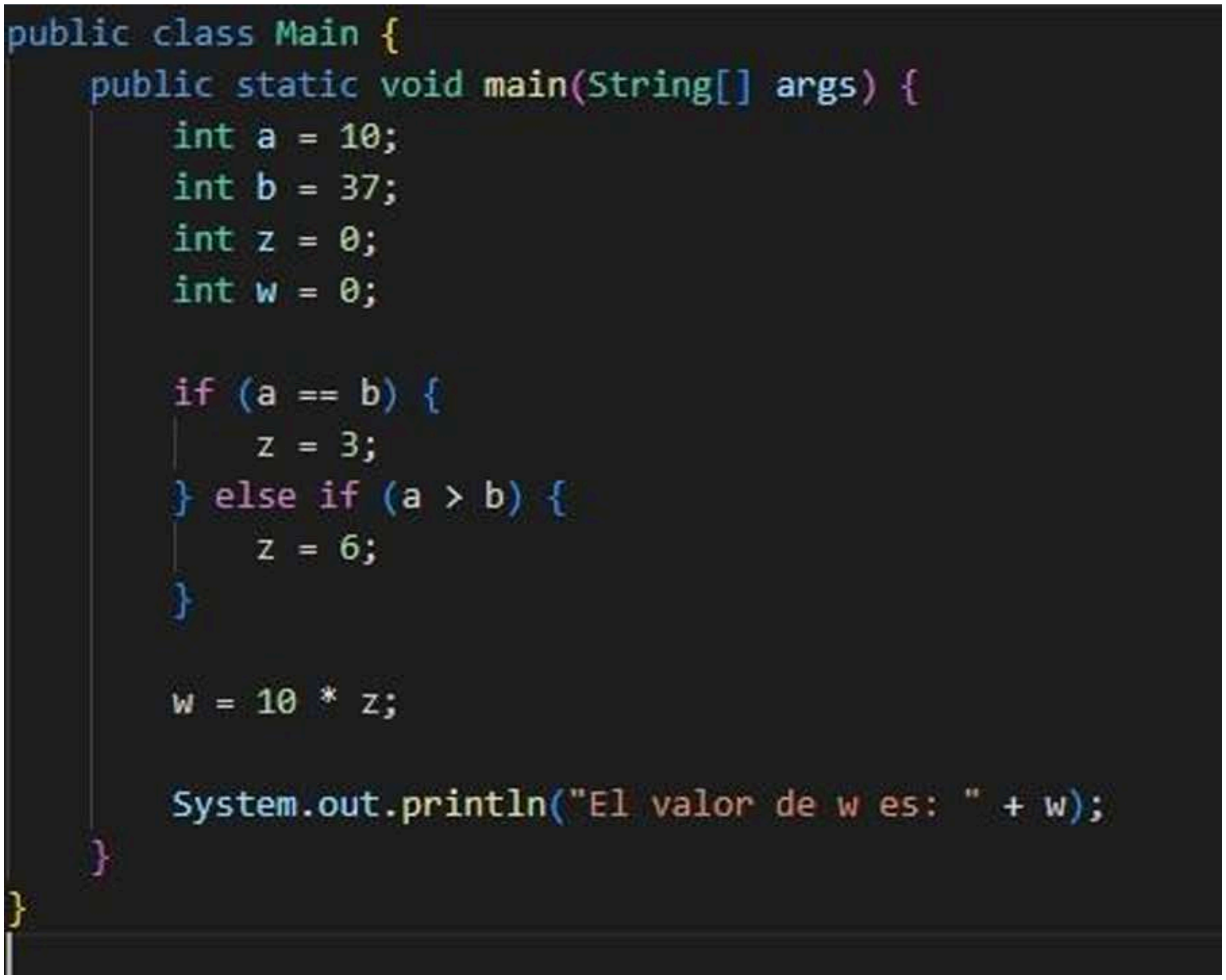
43. What is the result?



```
public static void main(String[]args){
    int[] array = { 1,2,3,4,5};
    System.arraycopy(array, 2, array, 1, 2);
    System.out.print(array[1]);
    System.out.print(array[4]);
}
```

A. 14
B. 15
C. 24
D. 25
E. 34
F. **35**

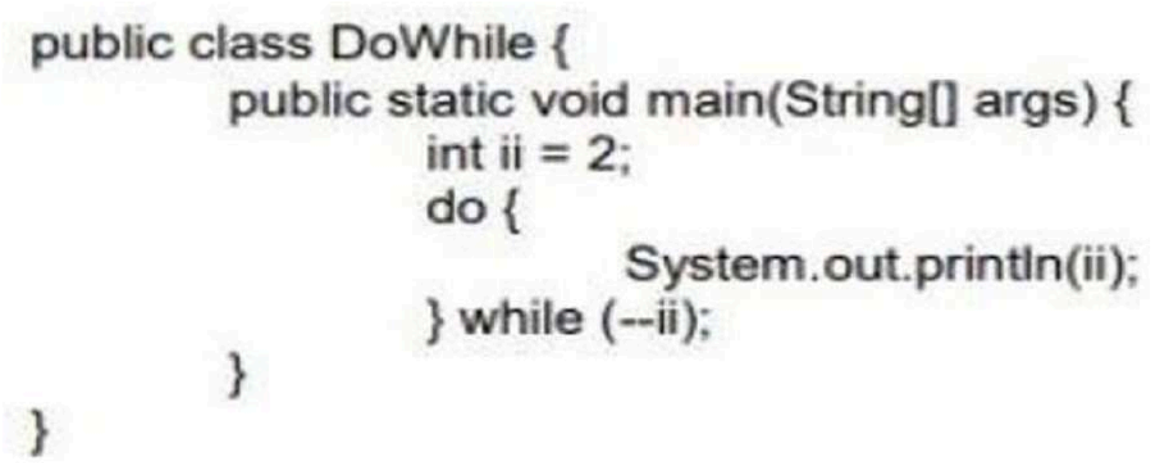
44. What is the result?



```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 37;
        int z = 0;
        int w = 0;

        if (a == b) {
            z = 3;
        } else if (a > b) {
            z = 6;
        }

        w = 10 * z;

        System.out.println("El valor de w es: " + w);
    }
}
```

A. **0**
B. 30
C. 60

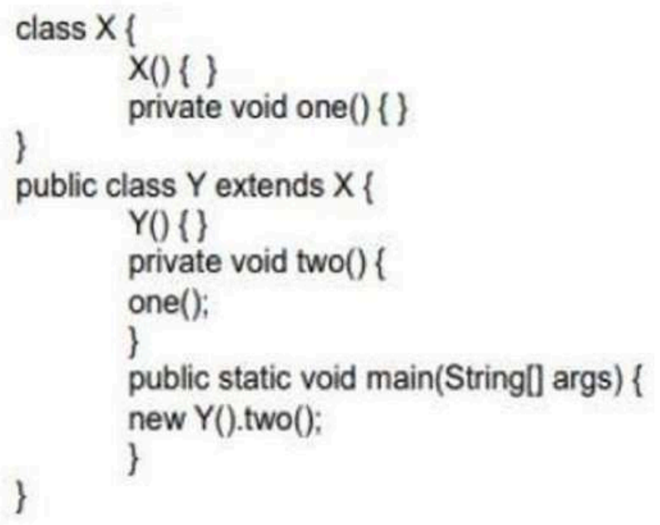
45. What is the result?



```
public class DoWhile {
        public static void main(String[] args) {
                int ii = 2;
                do {
                        System.out.println(ii);
                } while (--ii);
        }
}
```

A. 2 1 2
B. 1 0
C. null
D. An infinite loop.
E. Compilation fails

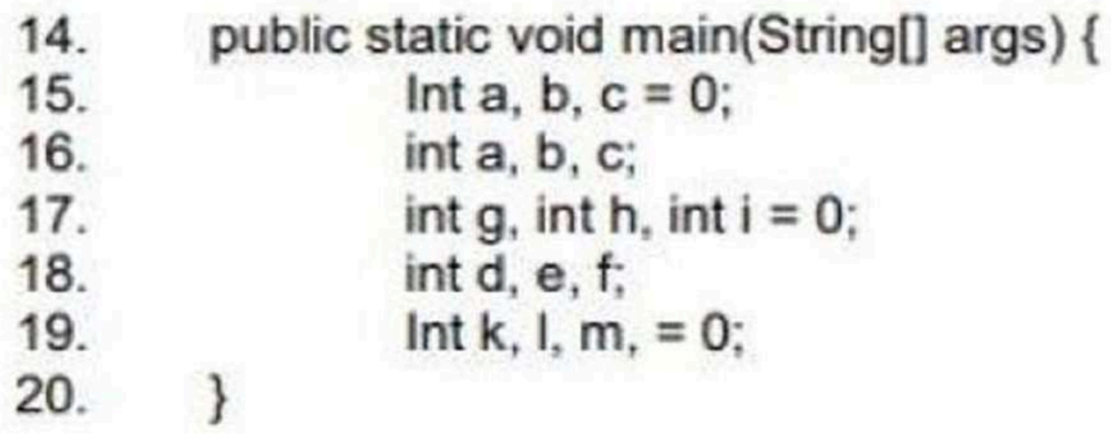
46. What changes will make this code compile?



```
class X {
        X() { }
        private void one() { }
}
public class Y extends X {
        Y() { }
        private void two() {
        one();
        }
        public static void main(String[] args) {
        new Y().two();
        }
}
```

A. Adding the public modifier to the declaration of class X.
B. Removing the Y() constructor.
C. Removing the private modifier from the two() method.
D. Adding the protected modifier to the X() constructor.
E. Changing the private modifier on the declarationof the one() method to protected.

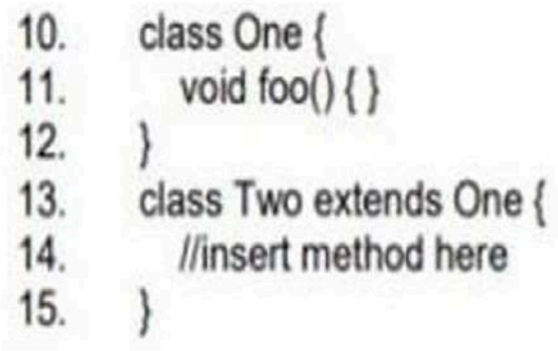
47. Which two declarations will compile?



```
14.     public static void main(String[] args) {
15.             Int a, b, c = 0;
16.             int a, b, c;
17.             int g, int h, int i = 0;
18.             int d, e, f;
19.             Int k, l, m, = 0;
20.     }
```

- Line 15.
- [✓ Line 16.]
- Line 17.
- [✓ Line 18.]
- Line 19
- Line 20

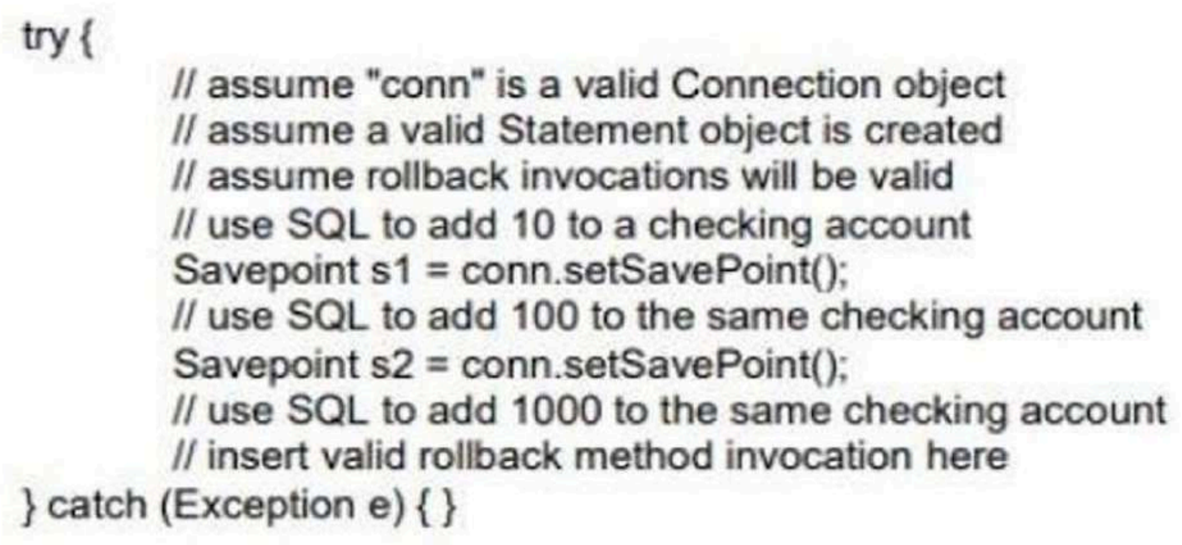
48. Which three methods, inserted individually at line, will correctly complete class Two(Choose three)?



```
10.   class One {
11.      void foo() { }
12.   }
13.   class Two extends One {
14.      //insert method here
15.   }
```

- [✓ public void foo() {/* more code here */}]
- private void foo() {/* more code here */}
- [✓ protected void foo() {/* more code here */}]
- [✓ int foo() {/* more code here */}]

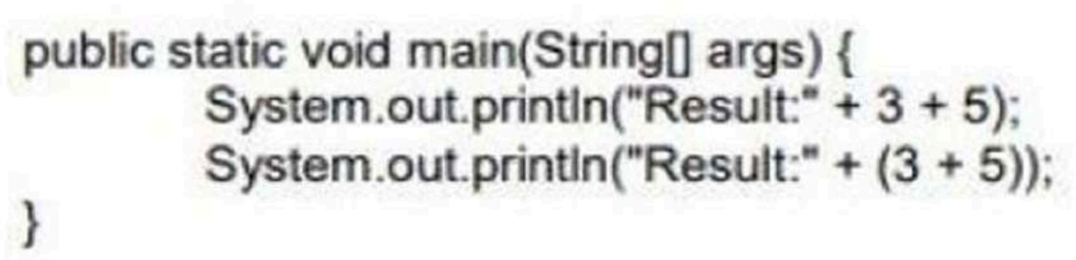
49. What is the result?



```
try {
        // assume "conn" is a valid Connection object
        // assume a valid Statement object is created
        // assume rollback invocations will be valid
        // use SQL to add 10 to a checking account
        Savepoint s1 = conn.setSavePoint();
        // use SQL to add 100 to the same checking account
        Savepoint s2 = conn.setSavePoint();
        // use SQL to add 1000 to the same checking account
        // insert valid rollback method invocation here
} catch (Exception e) { }
```

A. If conn.rollback(s1) is inserted, account will be incremented by 10.
B. If conn.rollback(s1) is inserted, account will be incremented by 1010.
C. If conn.rollback(s2) is inserted, account will be incremented by 100
D. If conn.rollback(s2) is inserted, account will be incremented by 110.
E. If conn.rollback(s2) is inserted, account will be incremented by 1110

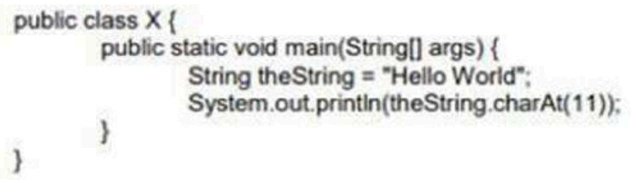
50. What is the result?



```
public static void main(String[] args) {
        System.out.println("Result:" + 3 + 5);
        System.out.println("Result:" + (3 + 5));
}
```

- Result: 8 Result: 8
- Result: 35 Result: 8
- Result: 8 Result: 35
- Result: 35 Result: 35

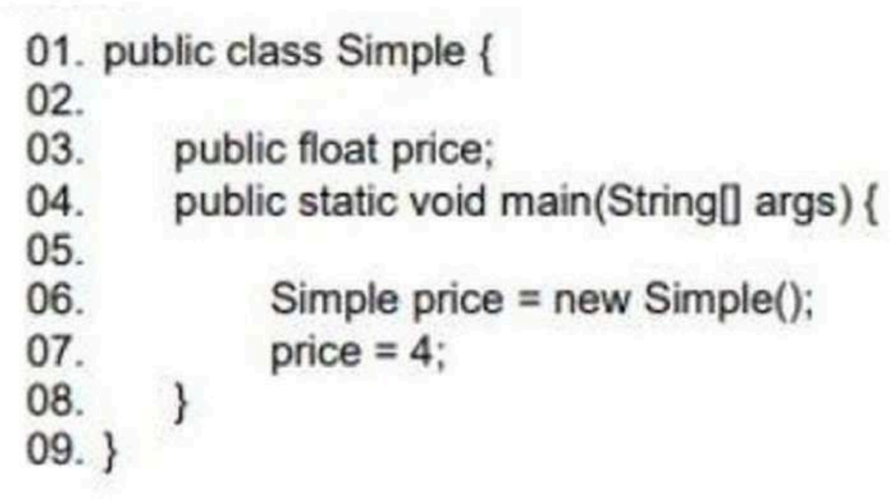
51. What is the result?



```
public class X {
        public static void main(String[] args) {
                String theString = "Hello World";
                System.out.println(theString.charAt(11));
        }
}
```

- There is no output.
- d is output.
- **A StringIndexOutOfBoundsException is thrown at runtime**.
- An ArrayIndexOutOfBoundsException is thrown at runtime.
- A NullPointException is thrown at runtime.
- A StringArrayIndexOutOfBoundsException is thrown at runtime.

52. What will make this code compile and run?



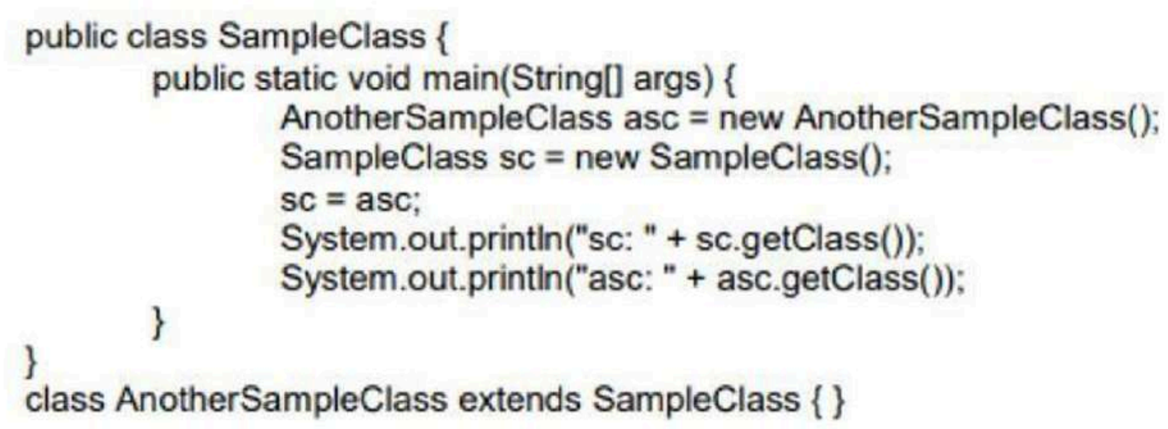
```
01. public class Simple {
02.
03.        public float price;
04.        public static void main(String[] args) {
05.
06.                Simple price = new Simple();
07.                price = 4;
08.        }
09. }
```

- **Change line 3 to the following: public int price**;
- Change line 7 to the following: int price = new Simple();
- Change line 7 to the following: float price = new Simple ();
- Change line 7 to the following: price = 4f;
- **Change line 7 to the following: price.price = 4**;

53. In the Java collections framework a Set is:

A. A collection that cannot contain duplicate elements.
B. An ordered collection that can contain duplicate elements
C. An object that maps value key sets and cannot contain values Duplicates

54. What is the result?



```
public class SampleClass {
        public static void main(String[] args) {
                AnotherSampleClass asc = new AnotherSampleClass();
                SampleClass sc = new SampleClass();
                sc = asc;
                System.out.println("sc: " + sc.getClass());
                System.out.println("asc: " + asc.getClass());
        }
}
class AnotherSampleClass extends SampleClass { }
```

A. sc: class.Object asc: class.AnotherSampleClass
B. sc: class.SampleClass asc: class.AnotherSampleClass
C. sc: class.AnotherSampleClass asc: class.SampleClass
D. sc: class.AnotherSampleClass asc: class.AnotherSampleClass

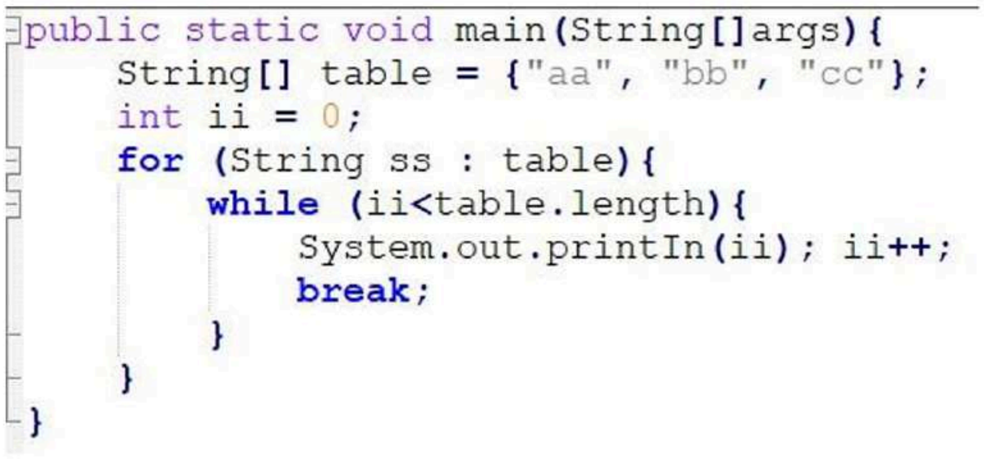
55. Which declaration initializes a boolean variable?

A. boolean j = (1 < 5);
B. boolean m = null;
C. boolean h = 1;
D. boolean k = 0;

56. Which two will compile, and can be run successfully usaing the following command?
Java Fred1 Hello walls.

- abstract class Fred1 {public static void main(String[] args){System.out.println(args[2]);}}
- class Fred 1{ public static void main(String args) { System.out.println(args[1]); } }
- class Fred1 { public static void main(String[] args) { System.out.println(args[1]); }
- class Fred1 { public static void main(String[] args) { System.out.println(args); } }

57. How many times is 2 printed?



```
public static void main(String[]args){
    String[] table = {"aa", "bb", "cc"};
    int ii = 0;
    for (String ss : table){
        while (ii<table.length){
            System.out.printIn(ii); ii++;
            break;
        }
    }
}
```

- Thrice
- It is not printed because compilation fails
- Twice
- Zero
- Once

58. Which two are valid instantiations and initializations of a multidimensional array?

A. **array2D[0][0] = 1; array2D[0][1] = 2; array2D[1][0] = 3; array2D[1][1] = 4;**

B. array3D[0][0] = array; array3D[0][1] = array; array3D[1][0] = array; array3D[0][1] =array;

C. int[][] array2D = {0, 1};

D. **int[][][] array3D = {{0, 1}, {2, 3}, {4, 5}}; int[] array = {0, 1}; int[][][] array3D = newint[2][2][2];**

E. int [][] array2D = {{0, 1, 2, 4}{5, 6}}; int[][] array2D = new int[2][2];

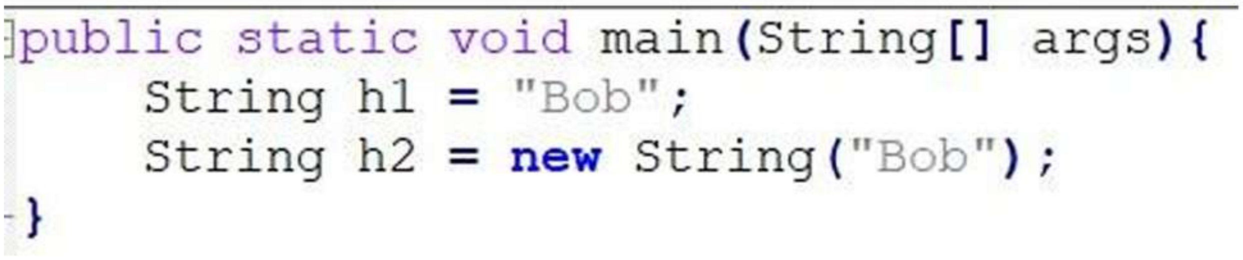
59. The standard API for accessing databases in Java is:

A. JPA/Hibernate

B. **JDBC**

C. ODBC

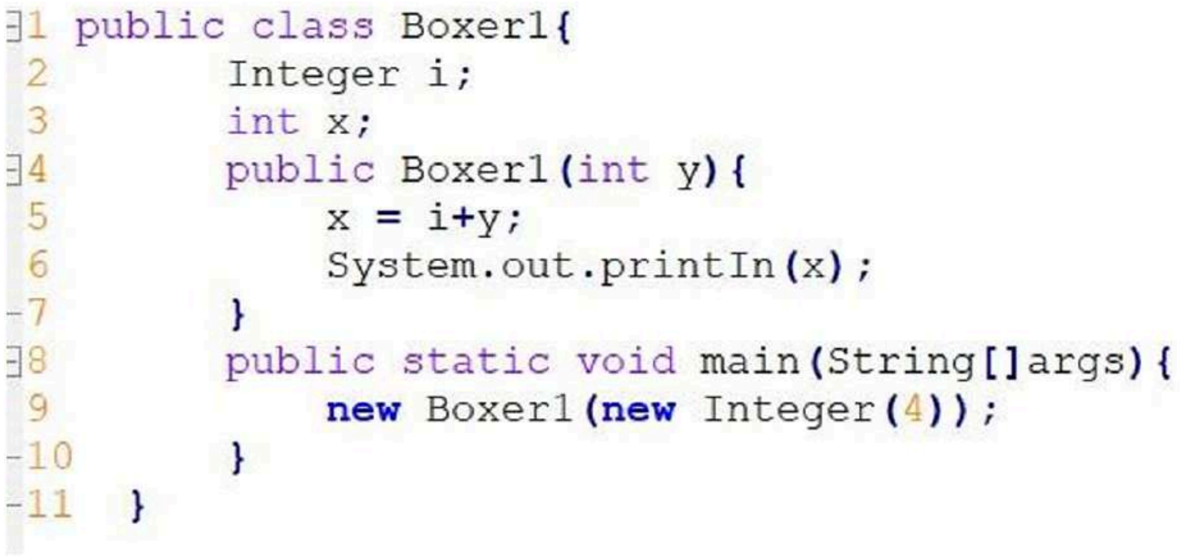
60. What is the best way to test that the values of h1 and h2 are the same?



```
public static void main(String[] args){
    String h1 = "Bob";
    String h2 = new String("Bob");
}
```

A. **if (h1.equals(h2)).**
B. if (h1 == h2).
C. if (h1.toString() == h2.toString()).
D. if (h1.same(h2)).

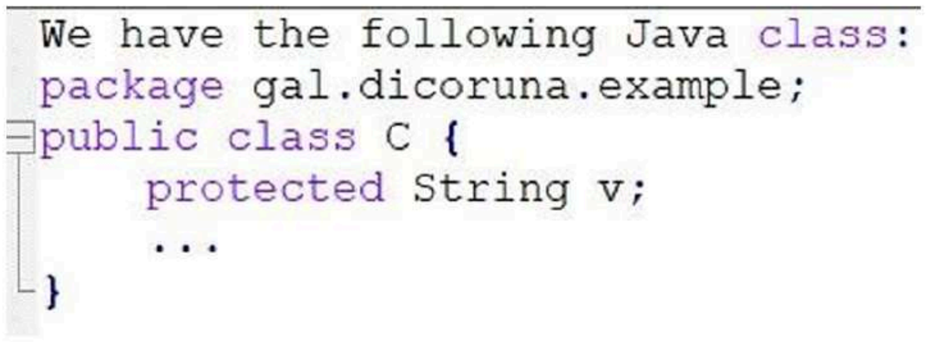
61. What is the result?



```
1  public class Boxer1{
2        Integer i;
3        int x;
4        public Boxer1(int y){
5            x = i+y;
6            System.out.printIn(x);
7        }
8        public static void main(String[]args){
9            new Boxer1(new Integer(4));
10       }
11   }
```

A. **A NullPointerException occurs at runtime.**
B. The value "4" is printed at the command line.
C. An IllegalStateException occurs at runtime.
D. Compilation fails because of an error in line 9.
E. Compilation fails because of an error in line 5.
F. A NumberFormatException occurs at runtime.

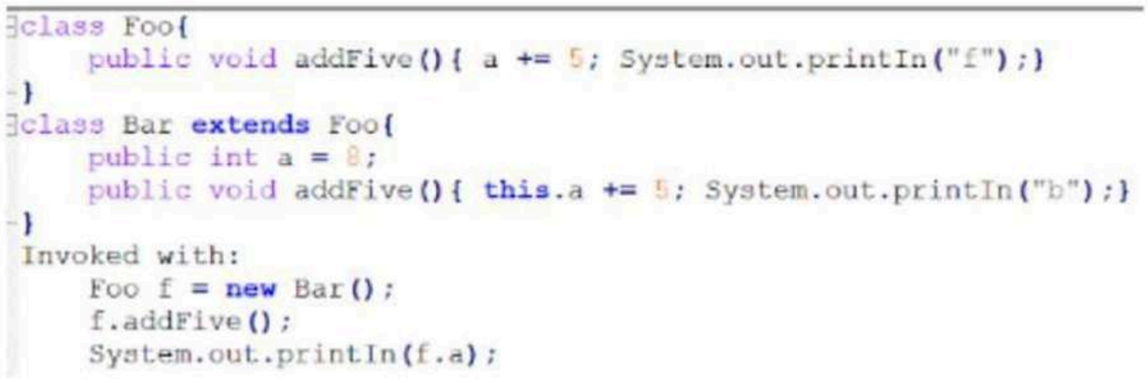
62. Given:



```
We have the following Java class:
package gal.dicoruna.example;
public class C {
    protected String v;
    ...
}
```

- From the class, the package, subclasses.
- Any site but only read if it is outside the package.
- From the class, the package, subclasses and all sites.

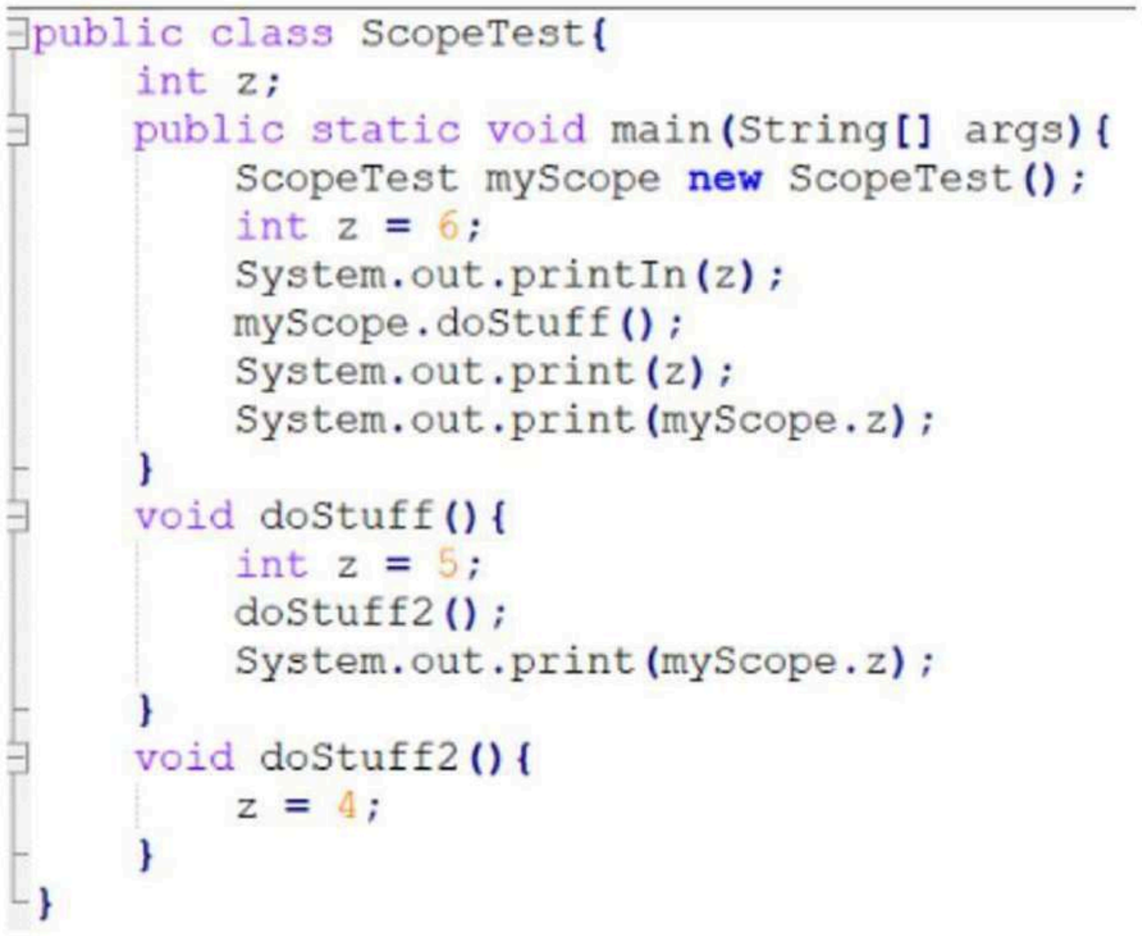
63. What is the result?



```
class Foo{
    public void addFive(){ a += 5; System.out.printIn("f");}
}
class Bar extends Foo{
    public int a = 8;
    public void addFive(){ this.a += 5; System.out.printIn("b");}
}
Invoked with:
    Foo f = new Bar();
    f.addFive();
    System.out.printIn(f.a);
```

- b 3
- Compilation fails
- f 8
- b 8
- f 13
- f 3
- An exception is thrown at runtime. b 13

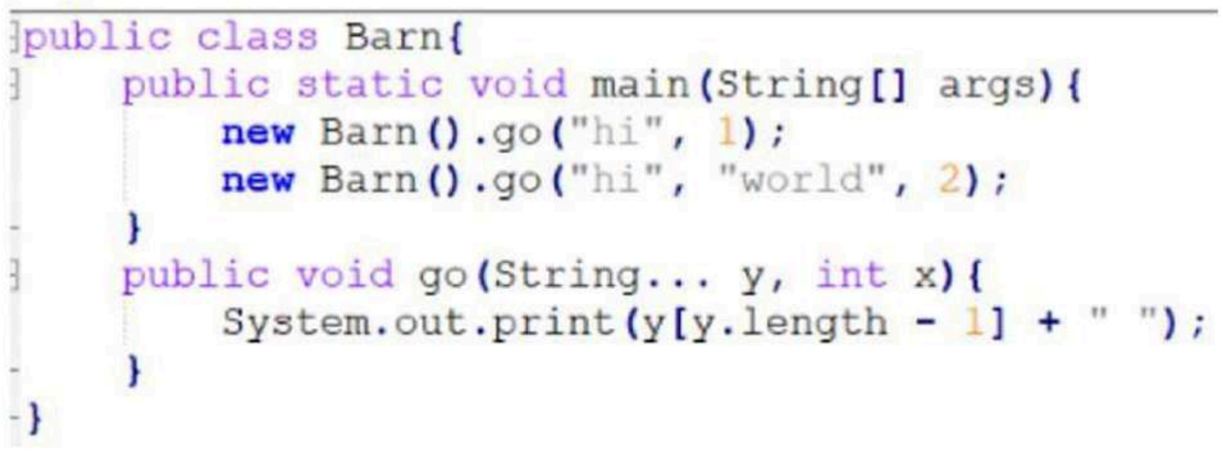
64. What is the result?



```
public class ScopeTest{
    int z;
    public static void main(String[] args){
        ScopeTest myScope new ScopeTest();
        int z = 6;
        System.out.printIn(z);
        myScope.doStuff();
        System.out.print(z);
        System.out.print(myScope.z);
    }
    void doStuff(){
        int z = 5;
        doStuff2();
        System.out.print(myScope.z);
    }
    void doStuff2(){
        z = 4;
    }
}
```

- 6554
- 6564
- 6565
- Compilation fails
- 6566

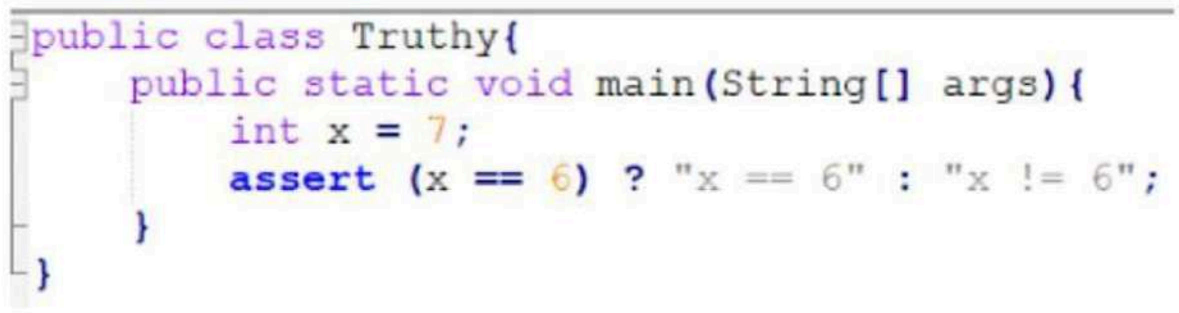
65. What is the result?



```
public class Barn{
    public static void main(String[] args){
        new Barn().go("hi", 1);
        new Barn().go("hi", "world", 2);
    }
    public void go(String... y, int x){
        System.out.print(y[y.length - 1] + " ");
    }
}
```

- worl world.
- hi hi
- hi world
- An exception is thrown at runtime.
- Compilation fails

66. What is the result if you try to compile Truthy.java and then run it with assertionsenabled?



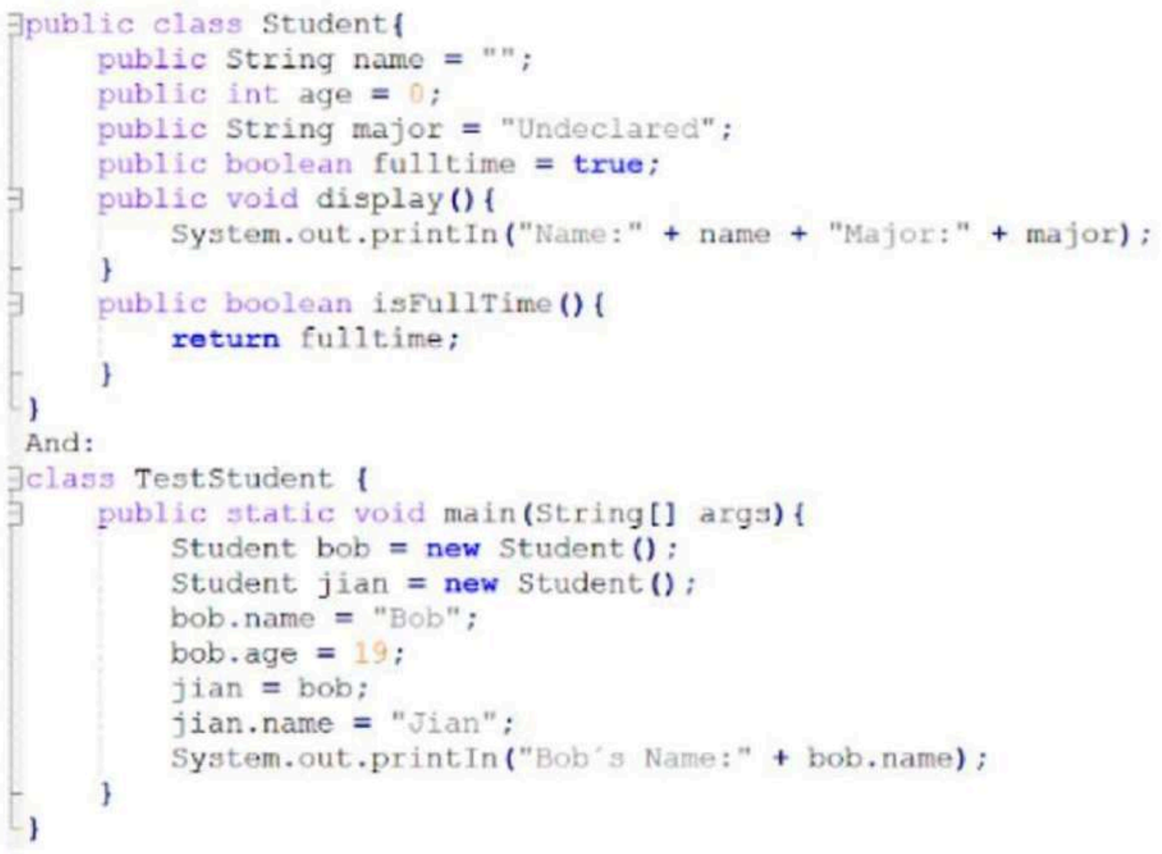
```
public class Truthy{
    public static void main(String[] args){
        int x = 7;
        assert (x == 6) ? "x == 6" : "x != 6";
    }
}
```

- Truthy.java compiles and the output is x != 6
- Truthy.java compiles and an AssertionError is thrown with x != 6 as additionaloutput.
- **Truthy.java does NOT compile.**
- Truthy.java compiles and an AssertionError is thrown with no additional output

67. What is the result? Given:

A. 0 tom 0 jerry 1 tom 1 jerry 2 tom 2 jerry
B. 0 tom 0 jerry 1 tom 2 tom 2 jerry
C. 0 tom 0 jerry 2 tom 2 jerry
D. 0 tom 0 jerry 1 tom 1 jerry

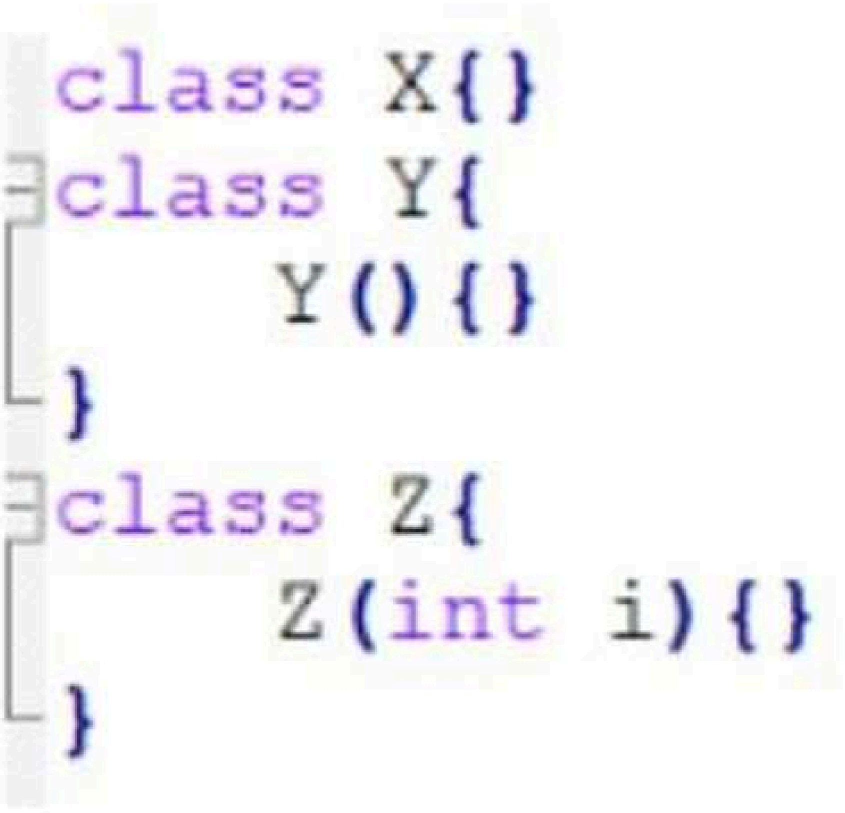
68. What is the result when this program is executed?



```
public class Student{
    public String name = "";
    public int age = 0;
    public String major = "Undeclared";
    public boolean fulltime = true;
    public void display(){
        System.out.printIn("Name:" + name + "Major:" + major);
    }
    public boolean isFullTime(){
        return fulltime;
    }
}
And:
class TestStudent {
    public static void main(String[] args){
        Student bob = new Student();
        Student jian = new Student();
        bob.name = "Bob";
        bob.age = 19;
        jian = bob;
        jian.name = "Jian";
        System.out.printIn("Bob's Name:" + bob.name);
    }
}
```

- Nothing prints
- Bob's name.
- Bob's Name: Bob.
- Bob's Name: Jian

69. Which class has a default constructor?



```
class X{}
class Y{
    Y(){}
}
class Z{
    Z(int i){}
}
```

- Z only.
- X only.
- X, Y and Z
- [✓ X and Y]
- X and Z
- Y only
- Y and Z

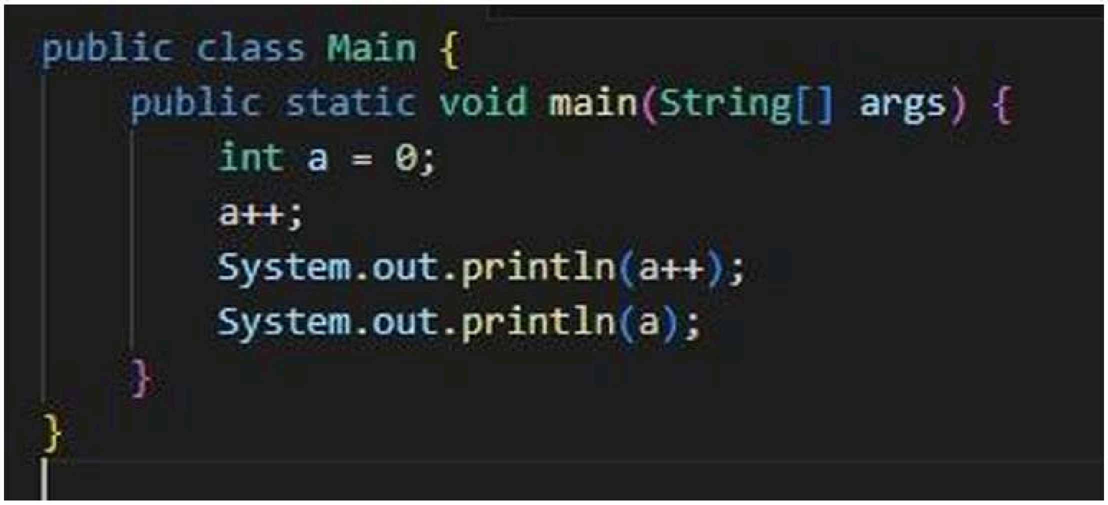
70. Which two may precede the word "class" in a class declaration?

A. **Public**
B. **Static**
C. Synchronized
D. Local
E. Volatile

71. The BUILDER pattern is used to:

A. Having several constructor methods in a class.
B. **Simplify the creation of complex objects.**
C. Implement the constructor of a class.

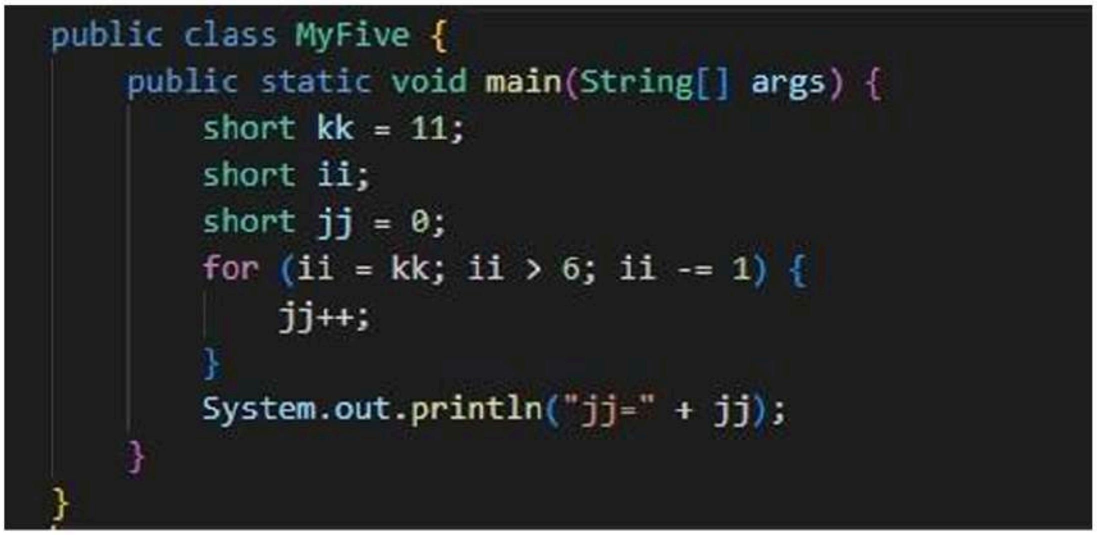
72. What is the result?



```java
public class Main {
    public static void main(String[] args) {
        int a = 0;
        a++;
        System.out.println(a++);
        System.out.println(a);
    }
}
```

- 0 1
- 1 2
- 2 2
- 1 1

73. What value should replace kk in the comment to cause jj = 5 to be output?



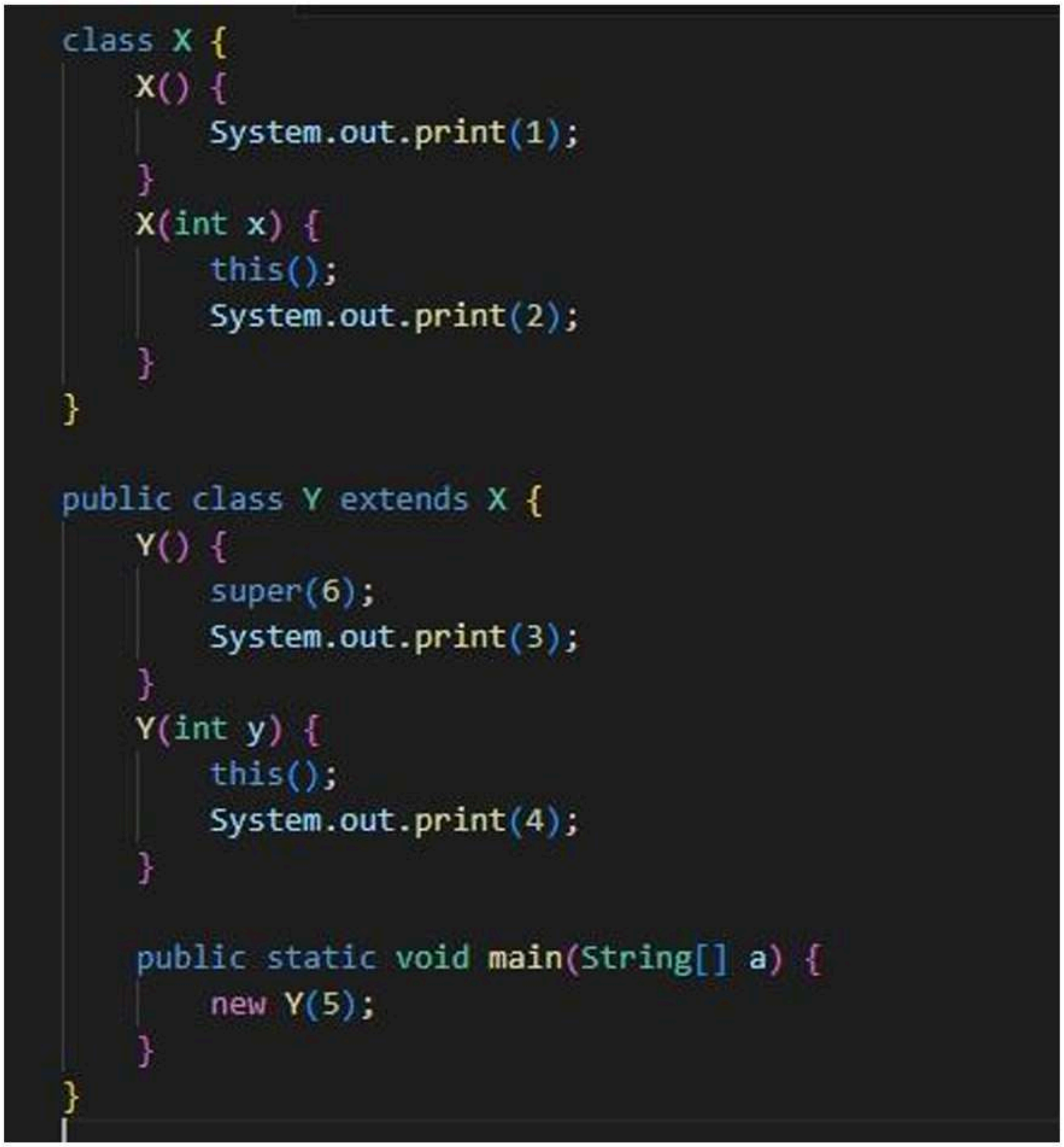
```
public class MyFive {
    public static void main(String[] args) {
        short kk = 11;
        short ii;
        short jj = 0;
        for (ii = kk; ii > 6; ii -= 1) {
            jj++;
        }
        System.out.println("jj=" + jj);
    }
}
```

- -1
- 1
- 5
- 8
- 11

74. What is a static block of code in Java?

A. A block of code within a class that runs whenever the class is load on the JVM.
B. A block of code inside a class that runs when that class first loaded in JVM
C. A block of code within a class that always runs before the builder.

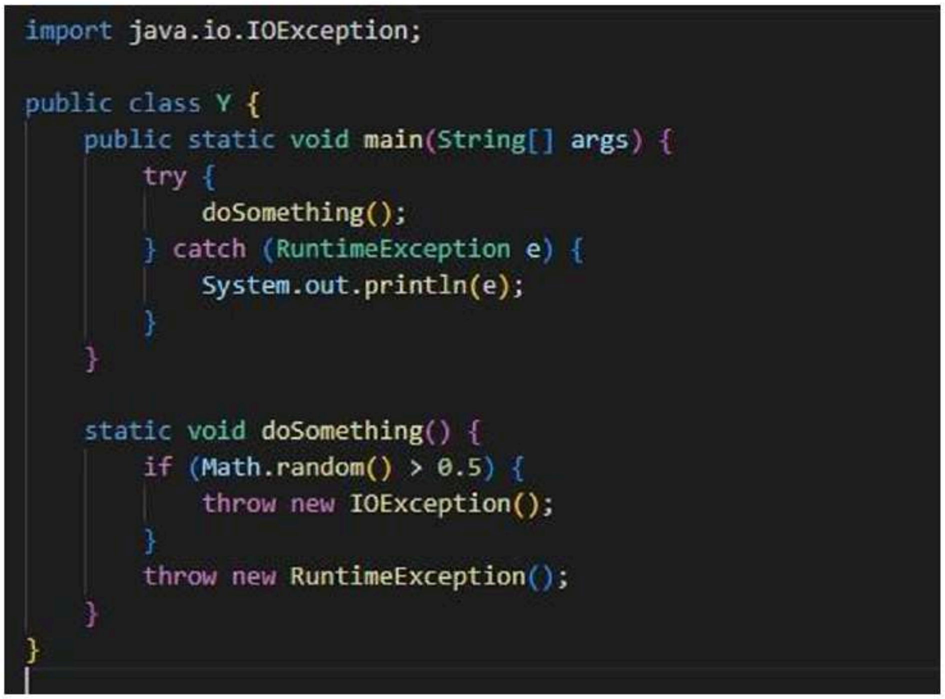
75. What is the result?



```java
class X {
    X() {
        System.out.print(1);
    }
    X(int x) {
        this();
        System.out.print(2);
    }
}

public class Y extends X {
    Y() {
        super(6);
        System.out.print(3);
    }
    Y(int y) {
        this();
        System.out.print(4);
    }

    public static void main(String[] a) {
        new Y(5);
    }
}
```

- 2134
- 4321
- 2143
- 13
- [✓ 1213]
- 134

76. Which two actions, used independently, will permit this class to compile?



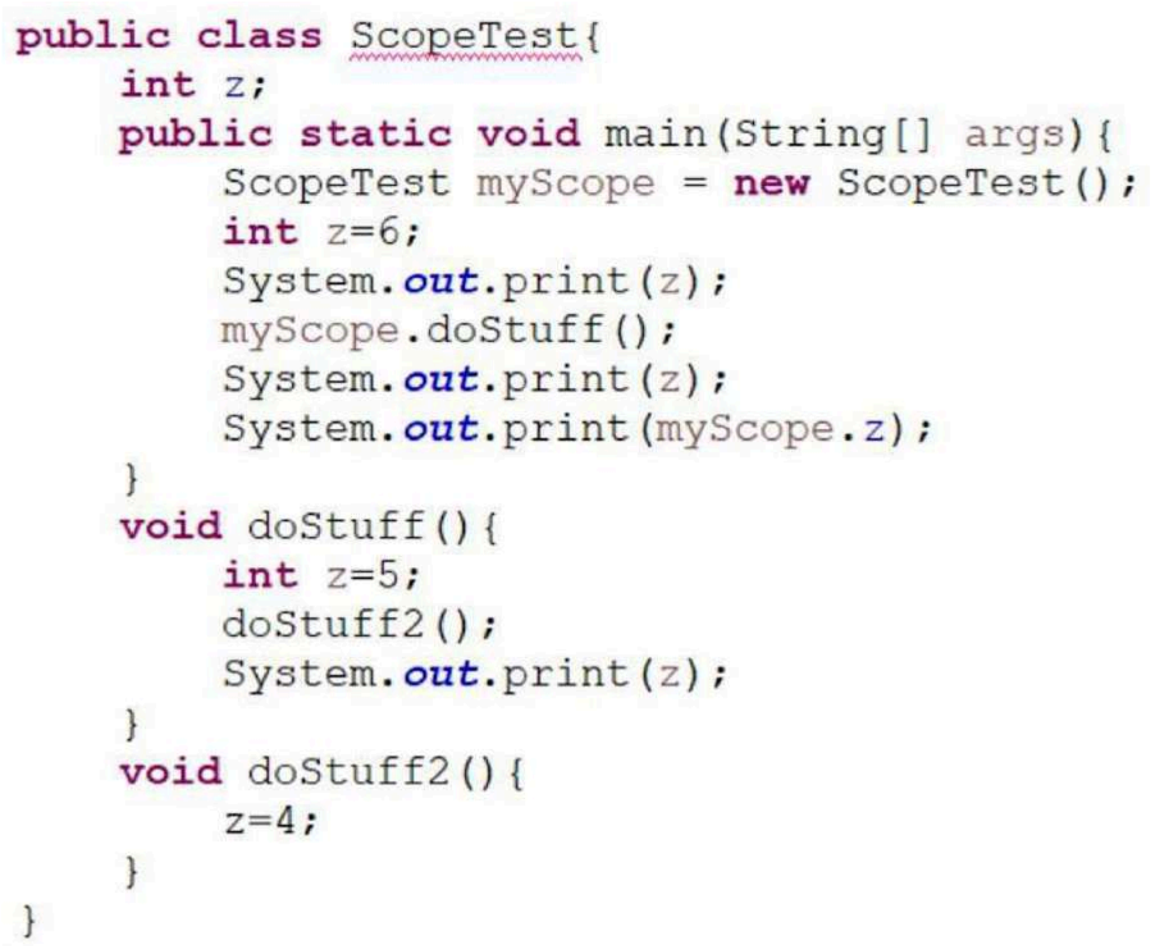
```
import java.io.IOException;

public class Y {
    public static void main(String[] args) {
        try {
            doSomething();
        } catch (RuntimeException e) {
            System.out.println(e);
        }
    }

    static void doSomething() {
        if (Math.random() > 0.5) {
            throw new IOException();
        }
        throw new RuntimeException();
    }
}
```

- Adding throws IOException to the main() method ....IOException
- Adding throws IOException to the dosomething() method signature and changing the catch
- Adding throws IOException to the main() method signature and to the dosomething() method
- Adding throws IOException to the main() method signature
- Adding throws IOException to the doSoomething() method signature.

77. A method is declared to take three arguments. A program calls this method andpasses only two arguments.What is the result?

A. Compilation fails
B. The third argument is given the value void.
C. The third argument is given the value zero
D. An exception occurs when the method attempts to access the third argument. The third argument is given the appropriate false value for its declared type. The third argument is given the value null.

78. What is the result?



```
public class ScopeTest{
    int z;
    public static void main(String[] args){
        ScopeTest myScope = new ScopeTest();
        int z=6;
        System.out.print(z);
        myScope.doStuff();
        System.out.print(z);
        System.out.print(myScope.z);
    }
    void doStuff(){
        int z=5;
        doStuff2();
        System.out.print(z);
    }
    void doStuff2(){
        z=4;
    }
}
```

- 6565
- 6504
- [✓ 6560]
- 6550

Cuestionario 2

1. ¿Qué es una API?
interfaz de programación de aplicaciones

2. ¿Qué es REST?
JavaScript Object Notation

3. ¿Que es un verbo HTTP?
HTTP define un conjunto de métodos o verbos de petición para indicar la acción que se desea realizar para un recurso determinado

4. Menciona al menos 3 verbos en HTTP
GET, POST, PUT

5. ¿En términos REST que significa idempotencia?
La ejecución repetida de una petición con los mismos parámetros sobre un mismo recurso tendrá el mismo efecto en el estado de nuestro recurso en el sistema si se ejecuta 1 o N veces

6. Menciona cuales son los verbos idempotentes
GET, PUT, DELETE

7. Menciona que verbo no es idempotente.
POST

8. Un verbo POST se utiliza para eliminar un recurso
Falso

9. Un verbo PATCH no se utiliza para actualizar un recurso de manera parcial.
Falso

10. Verbo HTTP que se utiliza para obtener/listar información
GET

11. ¿Qué es un código de retorno HTTP?
El código HTTP es en el que se indica si se ha completado satisfactoriamente o no una solicitud HTTP específica.

12. ¿Qué códigos de retorno HTTP se clasifican como informativos?
100

13. ¿Qué códigos de retorno HTTP se clasifican como de redireccionamiento?
300

14. ¿Qué códigos de retorno HTTP se clasifican como de respuesta satisfactoria?
200

15. ¿Qué códigos de retorno HTTP se clasifican como error de cliente?
400

16. ¿Qué códigos de retorno HTTP se clasifican como error del servidor?
500

17. Un código de error 400 se utiliza cuando hay un error en el servidor
Falso

18. Un código 201 se utiliza para devolver una respuesta satisfactoria pero con un contenido vacío.
Verdadero

19. ¿A qué le llamamos endpoint en un servicio REST?
Son las URLs de un API o un backend que responden a una petición.

20. ¿Cuál es la estructura correcta de un endpoint?
GET apiRest/v0/products}

21. ¿Qué es un Query Param en un servicio REST?
Son parámetros que son enviados al final del endpoint y siempre deberán estar después de un signo de interrogación.

22. ¿Qué es un UriParam en un servicio REST?
Parámetro de ruta, se usa básicamente para identificar un recurso o recursos específicos.

Cuestionario 2

Polimorfismo y Excepciones

1 Considera el siguiente bloque de código:

```
class Animal {

    void makeSound() throws Exception {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {

    void makeSound() throws RuntimeException {
        System.out.println("Dog barks");
    }
}

public class Main {

    public static void main(String[] args) {
        Animal myDog = new Dog();
        try {
            myDog.makeSound();
        } catch (Exception e) {
            System.out.println("Exception caught");
        }

}
```

¿Cuál sería la salida en consola al ejecutar este código?

1. Dog barks
2. Animal makes a sound
3. Exception caught
4. **Compilation error**

Hilos (Threads)
2 Considera el siguiente bloque de código:
```
class MyThread extends Thread {

    public void run() {
        System.out.println("Thread is running");
    }
}

public class Main {

    public static void main(String[] args) {
        Thread t1 = new MyThread();
        Thread t2 = new MyThread();
        t1.start();
        t2.start();
    }
}
```

¿Cuál sería la salida en consola al ejecutar este código?

1. Thread is running (impreso una vez)
2. Thread is running (impreso dos veces)
3. **Thread is running (imp res o dos veces , en orden aleator io)**
4. Compilation error

Listas y Excepciones
3 Considera el siguiente bloque de código:
```
import java.util.ArrayList;

import java.util.List;
public class Main {

    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>();
        numbers.add(1);
        numbers.add(2);
        numbers.add(3);
        try {
            for (int i = 0; i <= numbers.size(); i++) {
                System.out.println(numbers.get(i));
            }
        } catch (IndexOutOfBoundsException e) {
            System.out.println("Exception caught");
        }
    }
}
```
¿Cuál sería la salida en consola al ejecutar este código?

1. 1 2 3 Exception caught
2. 1 2 3
3. Exception caught
4. 1 2 3 4

Herencia, Clases Abstractas e Interfaces
4 Considera el siguiente bloque de código:
```
interface Movable {

    void move();
}

abstract class Vehicle {

    abstract void fuel();
}

class Car extends Vehicle implements Movable {

    void fuel() {
        System.out.println("Car is refueled");
    }

    public void move() {
        System.out.println("Car is moving");
    }
}

public class Main {

    public static void main(String[] args) {
        Vehicle myCar = new Car();
        myCar.fuel();
        ((Movable) myCar).move();
    }
}
```
¿Cuál sería la salida en consola al ejecutar este código?

1. Car is refueled Car is moving
2. Car is refueled
3. Compilation error

4. Runtime exception

Polimorfismo y Sobrecarga de Métodos
5 Considera el siguiente bloque de código:

```java
class Parent {

    void display(int num) {
        System.out.println("Parent: " + num);
    }

    void display(String msg) {
        System.out.println("Parent: " + msg);
    }
}

class Child extends Parent {

    void display(int num) {
        System.out.println("Child: " + num);
    }
}

public class Main {

    public static void main(String[] args) {
        Parent obj = new Child();
        obj.display(5);
        obj.display("Hello");
    }
}
```

¿Cuál sería la salida en consola al ejecutar este código?

1. Child: 5 Parent: Hello
2. Parent: 5 Parent: Hello
3. Child: 5 Child: Hello
4. Compilation error

Hilos y Sincronización
6 Considera el siguiente bloque de código:

```
class Counter {

    private int count = 0;

    public synchronized void increment() {
        count++;
    }

    public int getCount() {
        return count;
    }
}

class MyThread extends Thread {

    private Counter counter;

    public MyThread(Counter counter) {
        this.counter = counter;
    }

    public void run() {
        for (int i = 0; i < 1000; i++) {
            counter.increment();
        }
    }
}

public class Main {

    public static void main(String[] args) throws InterruptedException
{
        Counter counter = new Counter();
        Thread t1 = new MyThread(counter);
        Thread t2 = new MyThread(counter);
        t1.start();
        t2.start();
        t1.join();
        t2.join();
        System.out.println(counter.getCount());
    }
}
```

}
¿Cuál sería la salida en consola al ejecutar este código?

1. 2 00 0
2. 1000
3. Variable count is not synchronized
4. Compilation error

Listas y Polimorfismo
7 Considera el siguiente bloque de código:
```
import java.util.ArrayList;

import java.util.List;

class Animal {

    void makeSound() {
        System.out.println("Animal sound");
    }
}

class Dog extends Animal {

    void makeSound() {
        System.out.println("Bark");
    }
}

class Cat extends Animal {

    void makeSound() {
        System.out.println("Meow");
    }
}

public class Main {

    public static void main(String[] args) {
        List<Animal> animals = new ArrayList<>();
        animals.add(new Dog());
        animals.add(new Cat());
        animals.add(new Animal());

        for (Animal animal : animals) {
            animal.makeSound();
        }
    }
}
```

¿Cuál sería la salida en consola al ejecutar este código?

1. Animal sound Animal sound Animal sound
2. **Bark Meow Animal sound**
3. Animal sound Meow Bark
4. Compilation error

Manejo de Excepciones y Herencia
8 Considera el siguiente bloque de código:

```
class Base {

    void show() throws IOException {
        System.out.println("Base show");
    }
}

class Derived extends Base {

    void show() throws FileNotFoundException {
        System.out.println("Derived show");
    }
}

public class Main {

    public static void main(String[] args) {
        Base obj = new Derived();
        try {
            obj.show();
        } catch (IOException e) {
            System.out.println("Exception caught");
        }
    }
}
```

¿Cuál sería la salida en consola al ejecutar este código?
1. Base show
2. Derived show
3. Exception caught
4. **Compilation error**

Concurrencia y Sincronización

9 Considera el siguiente bloque de código:

```
class SharedResource {

    private int count = 0;

    public synchronized void increment() {
        count++;
    }

    public synchronized void decrement() {
        count--;
    }

    public int getCount() {
        return count;
    }
}

class IncrementThread extends Thread {

    private SharedResource resource;

    public IncrementThread(SharedResource resource) {
        this.resource = resource;
    }

    public void run() {
        for (int i = 0; i < 1000; i++) {
            resource.increment();
        }
    }
}

class DecrementThread extends Thread {

    private SharedResource resource;

    public DecrementThread(SharedResource resource) {
        this.resource = resource;

    }

    public void run() {
        for (int i = 0; i < 1000; i++) {
                resource.decrement();
            }
        }
    }

public class Main {

    public static void main(String[] args) throws InterruptedException
{
        SharedResource resource = new SharedResource();
        Thread t1 = new IncrementThread(resource);
        Thread t2 = new DecrementThread(resource);
        t1.start();
        t2.start();
        t1.join();
        t2.join();
        System.out.println(resource.getCount());
    }
}
```

¿Cuál sería la salida en consola al ejecutar este código?

1. 1000
2. 0
3. -1000
4. Compilation error

Generics y Excepciones
10 Considera el siguiente bloque de código:

```
class Box<T> {

    private T item;

    public void setItem(T item) {
        this.item = item;
    }

    public T getItem() throws ClassCastException {
        if (item instanceof String) {
            return (T) item; // Unsafe cast
        }
        throw new ClassCastException("Item is not a String");
    }
}

public class Main {

    public static void main(String[] args) {
        Box<String> stringBox = new Box<>();
        stringBox.setItem("Hello");
        try {
            String item = stringBox.getItem();
            System.out.println(item);
        } catch (ClassCastException e) {
            System.out.println("Exception caught");
        }
    }
}
```

¿Cuál sería la salida en consola al ejecutar este código?

1. Hello
2. Exception caught
3. Compilation error
4. ClassCastException

11 ¿Cuál es el resultado?
```
public class Main {

    public static void main(String[] args) {
        Padre objetoPadre = new Padre();
        Hija objetoHija = new Hija();
        Padre objetoHija2 = (Padre) new Hija();
        objetoPadre.llamarClase();
        objetoHija.llamarClase();
        objetoHija2.llamarClase();
        Hija objetoHija3 = (Hija) new Padre();
        objetoHija3.llamarClase();
    }
}

public class Hija extends Padre {

    public Hija() {
// Constructor de la clase Hija
    }

    @Override
    public void llamarClase() {
        System.out.println("Llame a la clase Hija");
    }
}

public class Padre {

    public Padre() {
// Constructor de la clase Padre
    }

    public void llamarClase() {
        System.out.println("Llame a la clase Padre");
    }
}
```

Elgie una:

1.
Llame a la clase Padre
Llame a la clase Hija
Llame a la clase Hija
Error: java. lang.Clas s CastException

2.

```
Llame a la clase Padre
Llame a la clase Hija
Llame a la clase Hija
Llame a la clase Hija
```

3.

```
Llame a la clase Padre
Llame a la clase Hija
Llame a la clase Hija
Llame a la clase Padre
```

12 ¿Cuál es la respuesta correcta?

```
import java.text.NumberFormat;
import java.text.ParseException;
import java.util.Scanner;
import java.util.ArrayList;

import java.util.List;
public class Ejemplos {

    public static void main(String[] args) {
        Animal uno = new Animal();
        Animal dos = new Dog();
        uno.makeSound();
        dos.makeSound();
        Dog tres = (Dog) new Animal();
        tres.makeSound();
    }
}

class Animal {

    void makeSound() {
        System.out.println("Animal sound");
    }
}

class Dog extends Animal {

    void makeSound() {
        System.out.println("Wau Wau");
    }
}
```

1. Animal sound Wau Wau compilation error
2. **Compilation Error**
3. Animal sound Wau Wau Animal sound
4. Animal sound

13 ¿Cuál es la respuesta correcta?

```
import java.text.NumberFormat;
import java.text.ParseException;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;

import java.lang.*;
public class Ejemplos {

    public static void main(String[] args) {
        Cambios uno = new Cambios();
        int x = 1;
        String hola = "hola";
        StringBuilder hola2 = new StringBuilder("hola2");
        Integer x2 = 4;
        uno.makeSound(x, hola);
        uno.makeSound(x2, hola2);
        System.out.println("Cambios?: " + x + "," + hola + "," + x2 +
"," + hola2);
    }
}

class Cambios {

    void makeSound(int x, String s) {
        s = "cambiando string";
        x = 5;
    }

    void makeSound(Integer x, StringBuilder s) {
        x = 9;
        s = s.delete(0, s.length());
    }
}
```

1. Compilation error
2. Cambios? : 1, hola, 4 ,
3. Cambios?: 1,hola,4,hola2
4. Cambios?: 5,cambiando string,9,

14 ¿Cuál es la respuesta correcta?

```
interface i1 {

    public void m1();
}

interface i2 extends i1 {

    public void m2();
}

class animal implements i1, i2 {
//¿Qué métodos debería implementar la clase animal en este espacio?
}
```

1. solo m1
2. **m1 y m2**
3. ninguno
4. error compilación

15 ¿Cuál es la respuesta correcta?

```
class Animal {

    void makeSound() throws Exception {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {

    void makeSound() throws RuntimeException {
        System.out.println("Dog barks");
    }
}

public class Main {

    public static void main(String[] args) {
        Animal myDog = new Dog();
        try {
            myDog.makeSound();
        } catch (Exception e) {
            System.out.println("Exception caught");
        }
    }
}
```

¿Cuál sería la salida en consola al ejecutar este código?

1. Dog barks
2. Animal makes a sound
3. Exception caught
4. Compilation error

16 ¿Cuál es la respuesta correcta?

```
import java.util.*;
import java.lang.*;

import java.io.*;
class Main {

    public static void main(String[] args) {
        String str = "1a2b3c4d5e6f";
        String[] splitStr = str.split("//D");
        for (String elemento : splitStr) {
            System.out.println(elemento);
        }
    }
}
```

Cuestionario 3

1. Which declaration initializes a boolean variable?

a) boolean m = null
[✓ b) Boolean j = (1<5)]
c) boolean k = 0
d) boolean h = 1

2. What is the DTO pattern used for?

[✓ e) To Exchange data between processes]
f) To implement the data Access layer
g) To implement the presentation layer

3. What value should replace kk in line 18 to cause jj = 5 to be output?

```
public class MyFive {
    public static void main(String[] args) {
        //short kk = ?;
        short ii; short
        jj = 0;
        for (ii = kk; ii > 6; ii-=1) {
               jj++;
        }
        System.out.println("jj = " + jj);
    }
}
```

a) -1
b) 1
c) 5
d) 8
e) 11

4 ¿Cuál será el resultado?

```
public class SampleClass {
   public static void main(String[] args) {
         SampleClass sc, scA, scB;
         sc = new SampleClass();
         scA = new SampleClassA();
         scB = new SampleClassB();
         System.out.println(“Hash is: “ + sc.getHash() +
         “, “ + scA.getHash() + “, “ + scB.getHash());
   }
   public int getHash() {
         return 111111;
   }
}
class SampleClassA extends SampleClass {
   public int getHash() {
         return 44444444;
   }
}
class SampleClassB extends SampleClass {
   public int getHash() {
         return 999999999;
   }
}
```

a) Compilation fails
b) An exception is thrown at runtime
c) There is no result because this is not correct way to determine the hash code
d) Hash is: 111111, 44444444, 999999999.

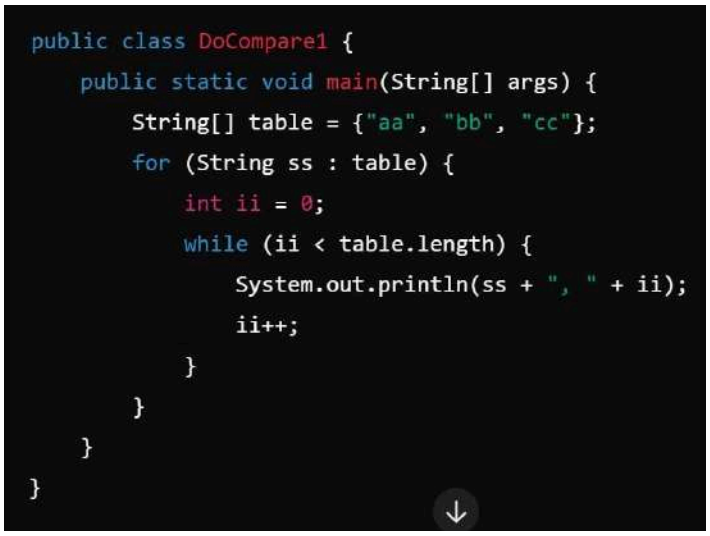
5. ¿Cuál sería el resultado?

```
public classDoCompare4 {
      public static void main(String[] args) {
            String[] table = {“aa”, “bb”, “cc”};
            int ii = 0;
            do {
                  while (ii < table.length) {
                        System.out.println(ii++);
                  }
            } while (ii < table.length);
```

```
        }
}
```

a) 0
b) 0 1 2
c) 0 1 2 0 1 2 0 1 2
d) Compilation fails

6 ¿Cuál sería el resultado?

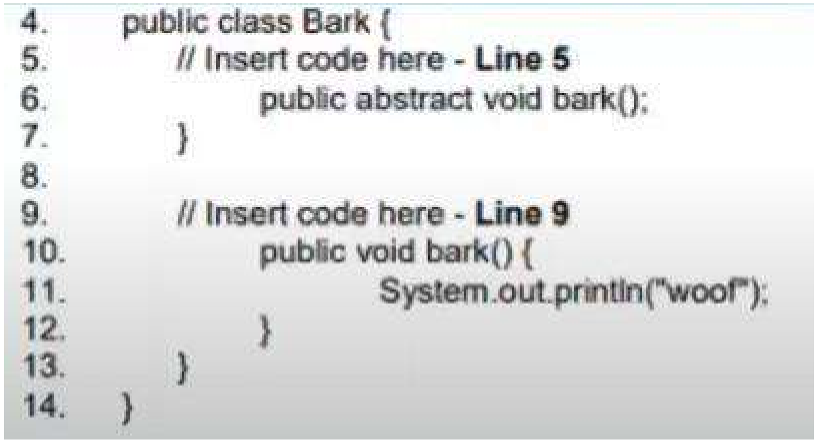

```
public class DoCompare1 {
    public static void main(String[] args) {
        String[] table = {"aa", "bb", "cc"};
        for (String ss : table) {
            int ii = 0;
            while (ii < table.length) {
                System.out.println(ss + ", " + ii);
                ii++;
            }
        }
    }
}
```

```
public class DoCompare1 {
    public static void main(String[] args) {
        String[] table = {"aa", "bb", "cc"};
        for (String ss : table) {
            int ii = 0;
            while (ii < table.length) {
                System.out.println(ss + ", " + ii);
                ii++;
            }
        }
    }
}
```

a) Zero.
b) Once.
c) Twice
d) Thrice

e) Compilation fails

7 What code should be inserted?



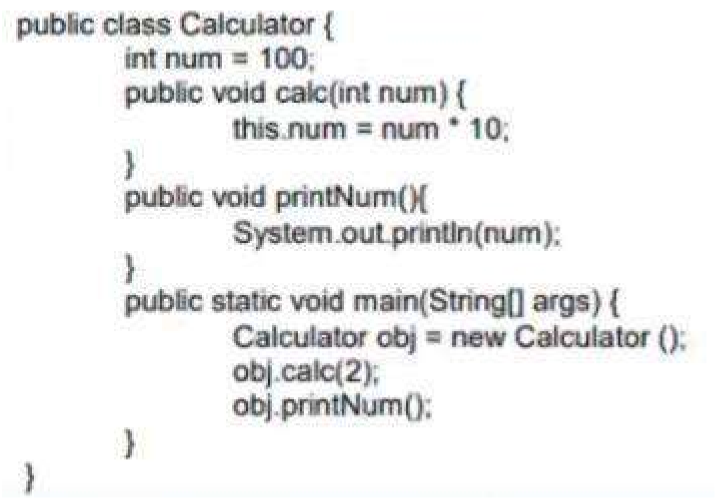
```
public class Bark {
    // Insert code here - Line 5
        public abstract void bark();
    }

    // Insert code here - Line 9
        public void bark() {
            System.out.println("woof");
        }
    }
}
```

a) 5. class Dog { 9. public class Poodle extends Dog {
b) 5. abstract Dog { 9. public class Poodle extends Dog {
c) 5. abstract class Dog { 9. public class Poodle extends Dog {
d) 5. abstract Dog { 9. public class Poodle implements Dog { e)
5. abstract Dog { 9. public class Poodle implements Dog {
f) 5. abstract class Dog { 9. public class Poodle implements Dog {

8 Wich statement initializes a stringBuilder to a capacity of 128?

a) StringBuilder sb = new String("128");
b) StringBuilder sb = StringBuilder.setCapacity(128); C.
c) StringBuilder sb = StringBuilder.getInstance(128); D.
d) StringBuilder sb = new StringBuilder (128);

9 What is the result?

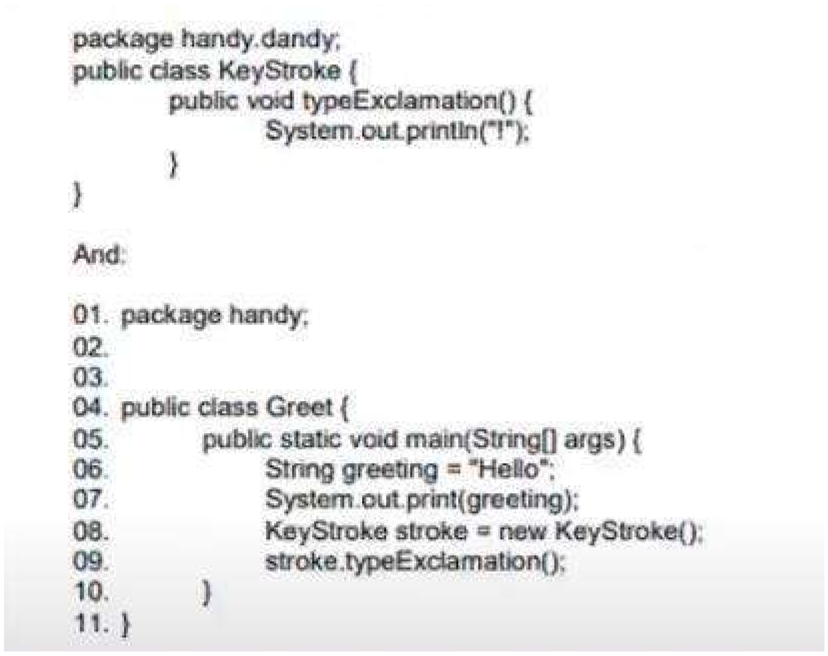


```
public class Calculator {
        int num = 100;
        public void calc(int num) {
                this.num = num * 10;
        }
        public void printNum(){
                System.out.println(num);
        }
        public static void main(String[] args) {
                Calculator obj = new Calculator ();
                obj.calc(2);
                obj.printNum();
        }
}
```

a) 20
b) 100
c) 1000
d) 2

.

10 What three modifications, made independently, made to class Greet, enable the code to compile and run?



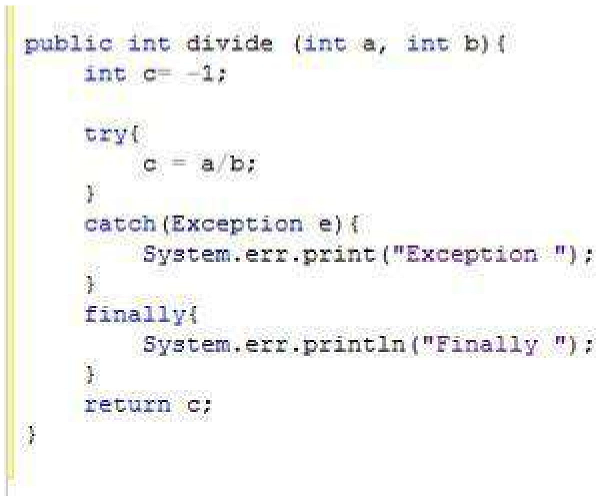
```
package handy.dandy;
public class KeyStroke {
        public void typeExclamation() {
                System.out.println("!");
        }
}
```

And:

```
01. package handy;
02.
03.
04. public class Greet {
05.         public static void main(String[] args) {
06.                 String greeting = "Hello";
07.                 System.out.print(greeting);
08.                 KeyStroke stroke = new KeyStroke();
09.                 stroke.typeExclamation();
10.         }
11. }
```

a) Line 8 replaced with handy.dandy.KeyStroke stroke = new KeyStroke();
b) Line 8 replaced with handy.*.KeyStroke stroke = new KeyStroke();
c) Line 8 replaced with handy.dandy.KeyStroke stroke = new handy.dandy.KeyStroke();
d) import handy.*; added before line 1.
e) import handy.dandy.*; added after line 1.
f) import handy.dandy.KeyStroke; added after line 1.
g) import handy.dandy.KeyStroke.typeExclamation(); added after line 1.

11 Consider the following Java code snippet:



```
public int divide (int a, int b){
    int c= -1;

    try{
        c = a/b;
    }
    catch(Exception e){
        System.err.print("Exception ");
    }
    finally{
        System.err.println("Finally ");
    }
    return c;
}
```

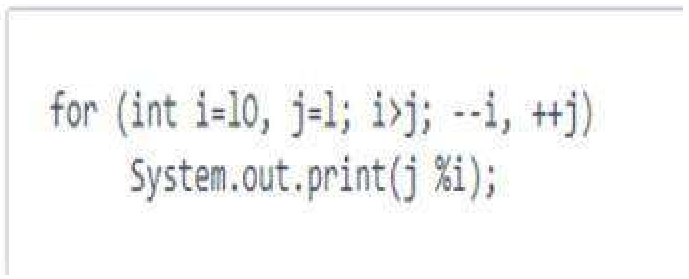
What will our code print when we call divide (4,0)?

a) [✓ Exception Finally]
b) Finally Exception
c) Exception

12 The feature which allows different methods to have the same name and arguments type, but the different implementation is called?

a) Overloading(SobreCarga)
b) [✓ Overriding (SobreEscritura @Override)]
c) Java does not permit methods with same and type signature
d) None of the above

13 What is the following for loop output?



```
for (int i=10, j=1; i>j; --i, ++j)
    System.out.print(j %i);
```

a) 12321
b) 12345
c) 11111
d) 00000

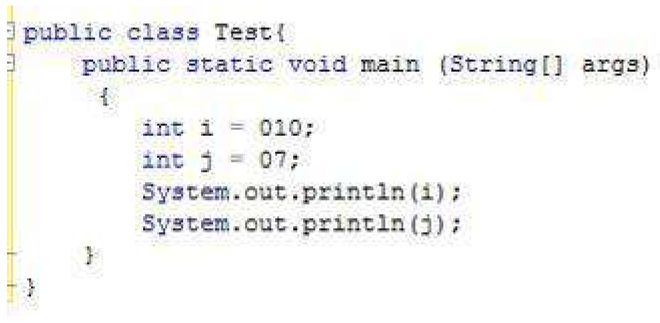
14 We perform the following sequence of actions:

1. Insert the following elements into a set: 1,2,9,1,2,3,1,4,1,5,7.
2. Convert the set into a list and sort it in ascending

order. Which option denotes the sorted list?

a) {1, 2, 3, 4, 5, 7, 9}
b) {9, 7, 5, 4, 3, 2, 1}
c) {1, 1, 1, 1, 2, 2, 3, 4, 5, 7, 9}
d) None of the above

15 What is the output for the Java code below?



```
public class Test{
    public static void main (String[] args)
     {
        int i = 010;
        int j = 07;
        System.out.println(i);
        System.out.println(j);
    }
}
```

a) 8 7
b) 10 7
c) Compilation fails with an error at line 3
d) Compilation fails with an error at line 5

16 A public data member with the same name is provided in both base as well as derived classes. Which of the following is true?

a) It is a compiler error to provide a field with the same name in both base and derived class

b) The program will compile and this feature is called overloading
c) The program will compile and this feature is called overriding
d) The program will compile and this feature is called as hiding or shadowing

17 Which statement is true?

a) Non-static member classes must have either default or public accessibility
b) All nested classes can declare static member classes
c) Methods in all nested classes can be declared static
d) Static member classes can contain non-static methods

18 A constructor is called whenever

a) An object is declared
b) An object is used
c) A class is declared
d) A class is used

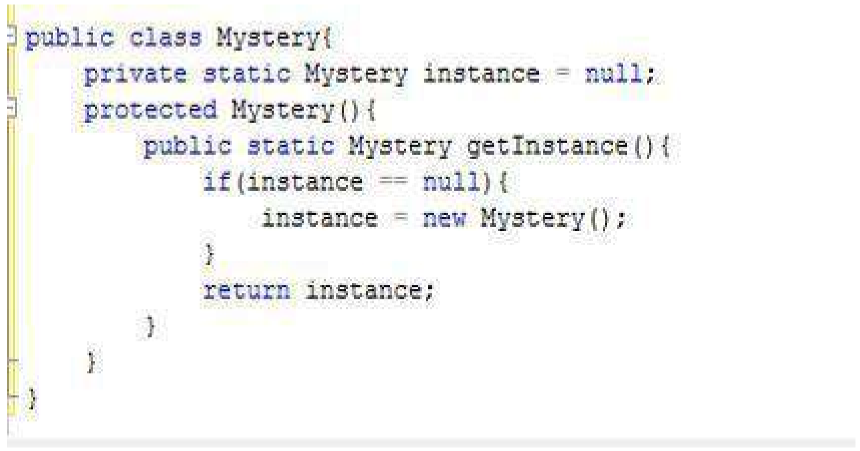
19 Which of the following data types in Java are primitive?

a) String
b) Struct
c) Boolean
d) Char

20 Which of the following are true for Java Classes?

a) The Void class extends the Class class
b) The Float class extends the Double class
c) The System class extends the Runtime class
d) The Integer class extends the Number class

21 The following code snippet is a demonstration of a particular design pattern. Which design pattern is it?



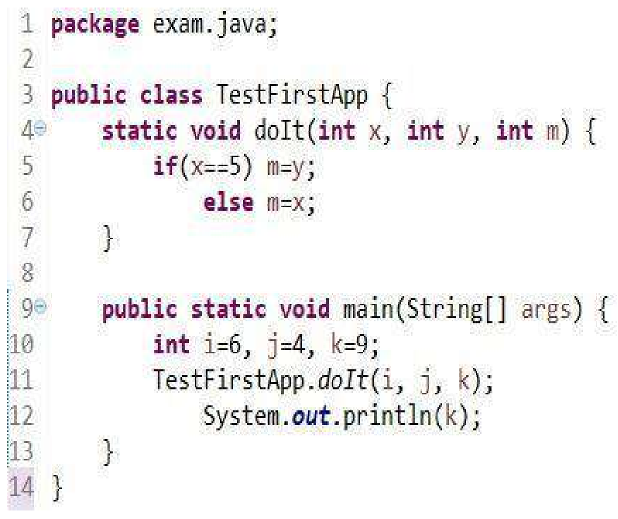
```
public class Mystery{
    private static Mystery instance = null;
    protected Mystery(){
        public static Mystery getInstance(){
            if(instance == null){
                instance = new Mystery();
            }
            return instance;
        }
    }
}
```

a) Factory Design Pattern
b) Strategy Pattern
c) Singleton
d) Facade Design Pattern

22 Which of the following Java declarations of the String array is correct?

a) String temp [] = new String {“j”, “a”, “z”};
b) String temp [] = {“j” “b” “c”};
c) String temp = {“a”, “b”, “c”};
d) S tring te mp [] = { “ a” , “b”, “c”} ;

23 Which is true of the following program?



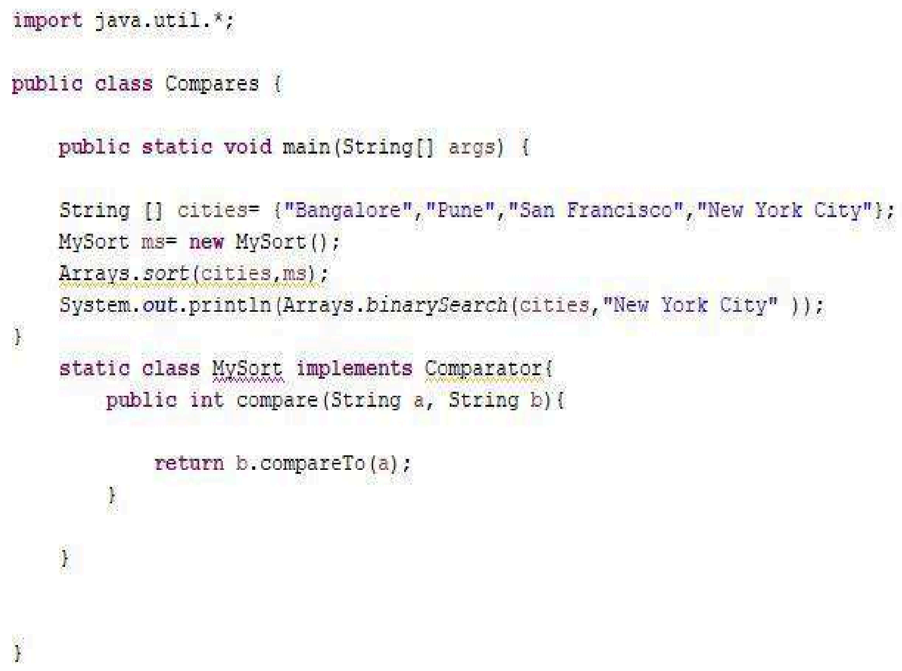
```
package exam.java;

public class TestFirstApp {
    static void doIt(int x, int y, int m) {
        if(x==5) m=y;
            else m=x;
    }

    public static void main(String[] args) {
        int i=6, j=4, k=9;
        TestFirstApp.doIt(i, j, k);
            System.out.println(k);
    }
}
```

a) Doesn’t matter what the values of *i* and *j* are, the output will always be *5*.
b) Doesn’t matter what the values of *k* and *j* are, the output will always be *5*.
c) Doesn’t matter what the values of *i* and *j* are, the output will always be *9*.
d) Doesn’ t matte r what the v alue s of *k* and *j* are, the output will
always be *9*.

24 Which of the following statements are correct. Select the correct answer.

a) Each Java file must have exactly one package statement to specify where the class is stored.
b) If a java file has both import and package statement, the import statement
must come before package statement.
c) A java file has at least one class defined
d) If a java file has a package statement, it must be the first statement (except comments).

25 Given the following code, what is the most likely result?



```
import java.util.*;

public class Compares {

    public static void main(String[] args) {

    String [] cities= {"Bangalore","Pune","San Francisco","New York City"};
    MySort ms= new MySort();
    Arrays.sort(cities,ms);
    System.out.println(Arrays.binarySearch(cities,"New York City" ));
}

    static class MySort implements Comparator{
        public int compare(String a, String b){

            return b.compareTo(a);
        }

    }


}
```

a) -
1 b)
1 c)
2
d) Compilation fails

26 To delete all pairs of keys and values in a given HashMap, which of the following methods should be used?

a) clearAll()
b) empty()
c) remove()
d) clear()

27 Which pattern do you see in the code below:

```
java.util.Calendar.getInstance();
```

a) Singleton Pattern
b) Factory Pattern
c) Facade Pattern
d) Adaptor Pattern

28 What is the output of the following program:

```
interface BaseI { void method ();
        } class BaseC
        {
                public void method()
                {
                        System.out.println( “I ns ide BaseC: :metho d”);
                }
        }
        class ImplC extends BaseC implements BaseI
        {
                public static void main (String []s)
                {
                        (new ImplC()).method();
                }
        }
```

a) Null
b) Complicati o fails
c) Inside BaseC::met h od
d) None of the above

29 Consider the following three classes:

```
class A {}
class B extends A
{} class C extends
B {}
```

Consider n object of class B is instantiated, i.e.,

```
B b = new B();
```

Which of the following Boolean expressions evaluates to true:

a) (b instanceof B)
b) (b instanceof B) && (!(b instanceof A))
c) (b instanceof B) && (!(b instanceof C))
d) None of the above

30 What is the output of the following program:

```
class Constructor
        {
        static String str;
        public void Constructor(){
                System.out.println("I n constructor" );
                str = " He llo Wor ld" ;
        }
        public static void main (String [] args){
                Constructor C = new Constructor
                (); System.out.println(str);
        }
}
```

a) In Constructor
b) Null
c) Compilation fails
d) None of the above

Cuestionario 4 Paola

1. "Given:

```
public class MyFor3 {
    public static void main(String_[] args) {
        int xx = null;
        System.out.println(xx);
    }
}
```

What is the result?"

A. null
B. compilation fails
C. Java.lang. NullPointerException
D. o

2. Given a java source file

```
class X{
    X(){}
    private void one(){}
}
public class Y extends X {
    Y(){}
    private void two(){
        one();
    }
    public static void main(String[] args){
        new Y().two();
    }
}
```

What changes will make this code compile?

A. adding the public modifier to the declaration of class X
B. adding the protected modifier to the X) constructor
C. changing the private modifier on the declaration of the one() method to protected
D. removing the Y () constructor

8. "Given the code fragment:
String h1 = "Bob";
String h2 = new String ("Bob");
What is the best way to test that the values of hi and h2 are the same?"

A. if (h1 = = h2)
B. (h1.equals(h2))
C. if (h1 = = h2)
D. if (h1.same(h2))

9. Given the code fragment

```
class Switch {
    void metodo() {
        String color = "Red";
        switch (color) {
            case "Red":
                System.out.println("Found Red");
            case "Blue":
                System.out.println("Found Blue");
            case "White":
                System.out.println("Found White");
                break;
            default:
                System.out.println("Found Default");
        }
    }
}
```

What is the result?

A. Found Red
B. Found Red Found Blue
C. Found Red Found Blue Found White
D. Found Red Found Blue Found White Found Default

10. Given

```
public class Bark {
   // Insert code here - Line 5

    public abstract void bark(); // Line 6
    // Line 7
    // Line 8
    // Insert code here - Line 9

    public void bark() {
        System.out.println("woof");
    }
}
```

What code should be inserted?"

A. 5. class Dog { 9. public class Poodle extends Dog {"
B. 5. abstract Dog { 9. public class poodle extends Dog {"
C. 5. abstract class Dog { 9. public class Poodle extends Dog {"
D. 5. abstract Dog { 9. public class Poodle implements Dog {"

12 Given the code fragment

```
public class Ejercicio {
    void metodo(){
        int j = 0, k = 0;
        for (int i=0; i<x; i++){
            do{
                k=0;
                while(k<z){
                    k++;
                    System.out.println(k+" ");
                }
                System.out.println(" ");
                j++;
            }while(j<y);
            System.out.println("--------");
        }
    }
}
```

What values of x, Y, z will produce the following result?

```
 1 2 3 4
 1 2 3 4
 1 2 3 4
 --------
 1 2 3 4
 --------
```

A. X = 2, Y = 3, Z = 4
B. X = 3, Y = 2, Z = 3
C. X = 2, Y = 3, Z = 3
D. X = 4, Y = 2, 2 = 3

12 Given

```
class X{}
class Y{(){}}
class Z{Z(int i){}}
```

Which class has a default constructor?

A. X only
B. X and Z
C. Z only
D. X and Y

13 Given

```
class Overloading {

    int (double d) {
        System.out.println("one");
        return 0;
    }

    String x(double d) {
        System.out.println("two");
        return null;
    }

    double x(double d) {
        System.out.println("three");
        return 0.0;
    }

    public static void main(String[] args) {
        new Overloading().x(4.0);
    }
}
```

What is the result?

A. One
B. Two
C. Three
D. Compilation fails

14 Given

```
public class X implements Z {

    public String toString() {
        return "X";
    }

    public static void main(String[] args) {
        Y myY = new Y();
        X myX = myY;
        Z myZ = myX;
        System.out.print(myX);
        System.out.print((Y) myX);
        System.out.print(myZ);
    }
}

class Y extends X {

    public String toString() {
        return "Y";
    }
}

interface Z {
}
```

A. X X X
B. X Y X
C. Y Y X
D. Y Y Y

15. Given the code fragment

```
int[][] array = {{0},{0,1},{0,2,4},{0,3,6,9},{0,4,8,12,16}};
System.out.println(array[4][1]);
System.out.println(array[1][4]);
```

A. 4 Null
B. Null 4
C. An IllegalArgumentException is thrown at run time
E. 4 An ArrayIndexOutOfBoundException is thrown at run time

16 Given

```
public class MyFor {

    public static void main(String_[] args) {
        for (int ii = 0; ii < 4; ii++) {
            System.out.println("i = " + ii);
            ii = ii + 1;
        }
    }
}
```

What is the result?"

A. ii = 0 ii = 2
B. ii = 0 ii = 1 ii = 2 ii = 3
C. ii =
D. Compilation fails.

17 Given

```
public class Message {

    public static void main(String[] args) {
        String message1 = "Wham bam!";
        String message2 = new String("Wham bam!");
        if (message1 == message2)
            System.out.println("They match");
        if (message1.equals(message2))
            System.out.println("They really match");
    }
}
```

A. They match They really match"
B. They really match
C. They match
D. Nothing Prints

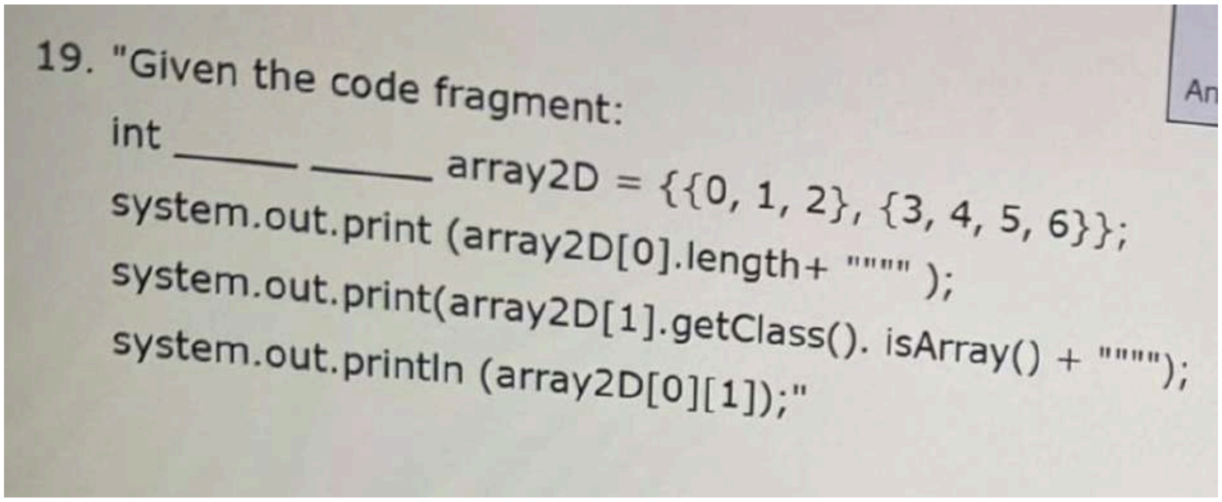
18 Given

```
public class SampleClass {

    public static void main(String[] args) {
        AnotherSampleClass asc = new AnotherSampleClass();
        SampleClass sc = new SampleClass();
        // TODO code application logic here
    }
}

class AnotherSampleClass extends SampleClass {
}
```

Which statement, when inserted into line ""// TODO code application logic here", is valid Change?

A. asc = sc;
B. sc = asc;
C. asc = (object) sc;
D. asc = sc.clone ()

19 Given the code fragment

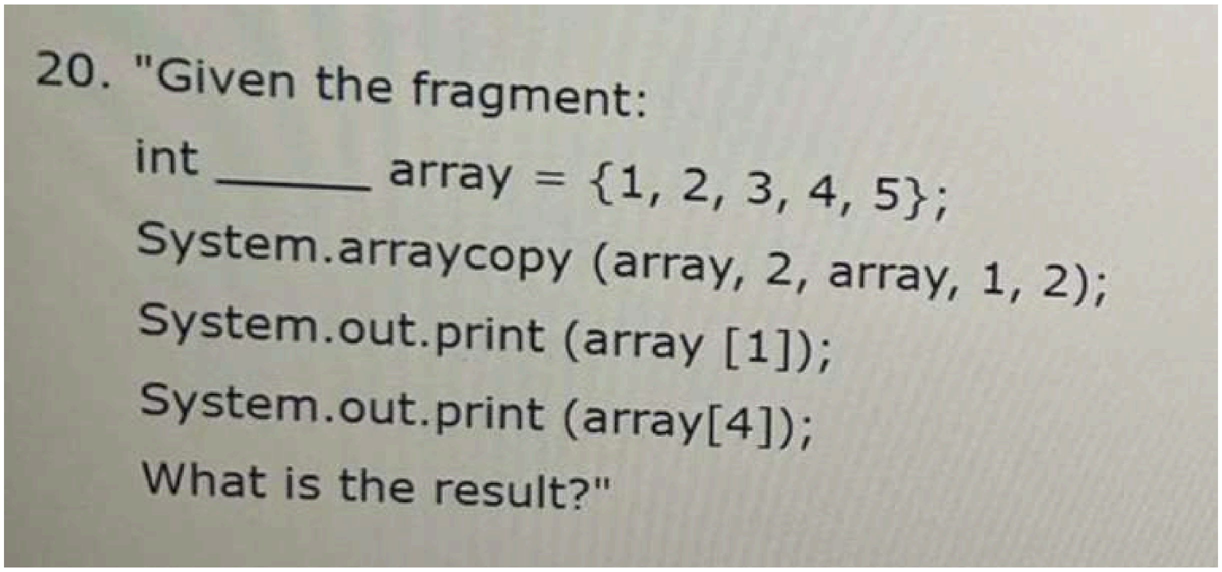


```
19. "Given the code fragment:
    int ______ ______ array2D = {{0, 1, 2}, {3, 4, 5, 6}};
    system.out.print (array2D[0].length+ """" );
    system.out.print(array2D[1].getClass(). isArray() + """");
    system.out.println (array2D[0][1]);"
```

An

A. 3false1"
B. 2true3
C. 2false3
D. 3true1

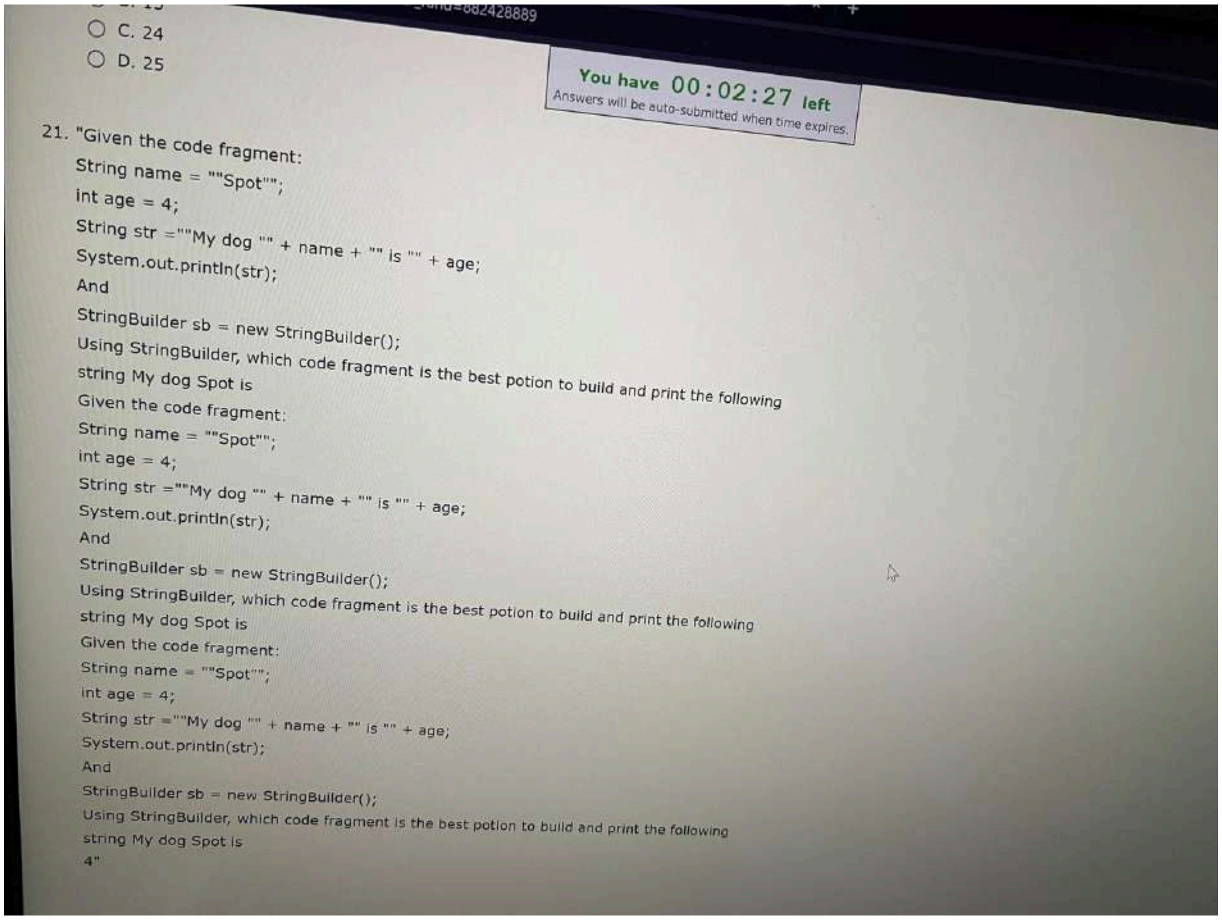
20. Given the fragment



```
20. "Given the fragment:
    int ______ array = {1, 2, 3, 4, 5};
    System.arraycopy (array, 2, array, 1, 2);
    System.out.print (array [1]);
    System.out.print (array[4]);
    What is the result?"
```

A. 35
B. 15
C. 24
D. 25

21. Given the code fragment



C. 24

D. 25

You have 00:02:27 left
Answers will be auto-submitted when time expires.

21. "Given the code fragment:

```
String name = ""Spot"";
int age = 4;
String str =""My dog "" + name + "" is "" + age;
System.out.println(str);
```

And

```
StringBuilder sb = new StringBuilder();
```

Using StringBuilder, which code fragment is the best potion to build and print the following string My dog Spot is

Given the code fragment:

```
String name = ""Spot"";
int age = 4;
String str =""My dog "" + name + "" is "" + age;
System.out.println(str);
```

And

```
StringBuilder sb = new StringBuilder();
```

Using StringBuilder, which code fragment is the best potion to build and print the following string My dog Spot is

Given the code fragment:

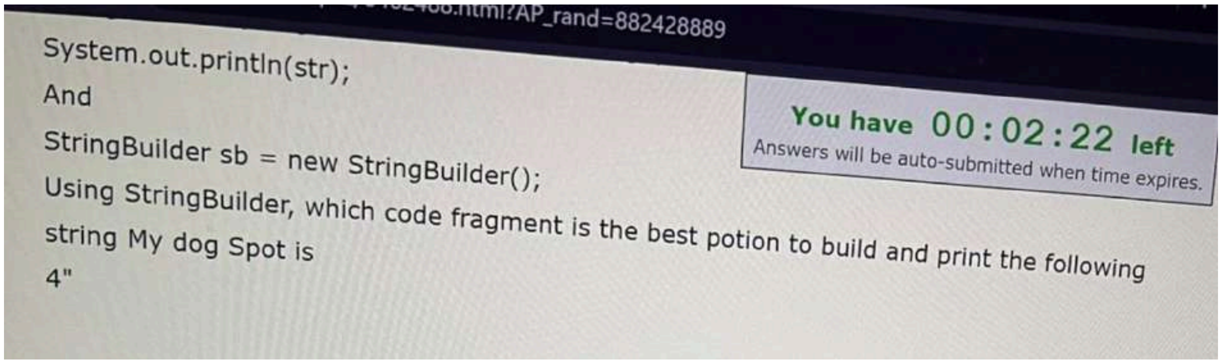
```
String name = ""Spot"";
int age = 4;
String str =""My dog "" + name + "" is "" + age;
System.out.println(str);
```

And

```
StringBuilder sb = new StringBuilder();
```

Using StringBuilder, which code fragment is the best potion to build and print the following string My dog Spot is

4"



```
System.out.println(str);
```

And

```
StringBuilder sb = new StringBuilder();
```

Using StringBuilder, which code fragment is the best potion to build and print the following string My dog Spot is

4"

You have 00:02:22 left
Answers will be auto-submitted when time expires.

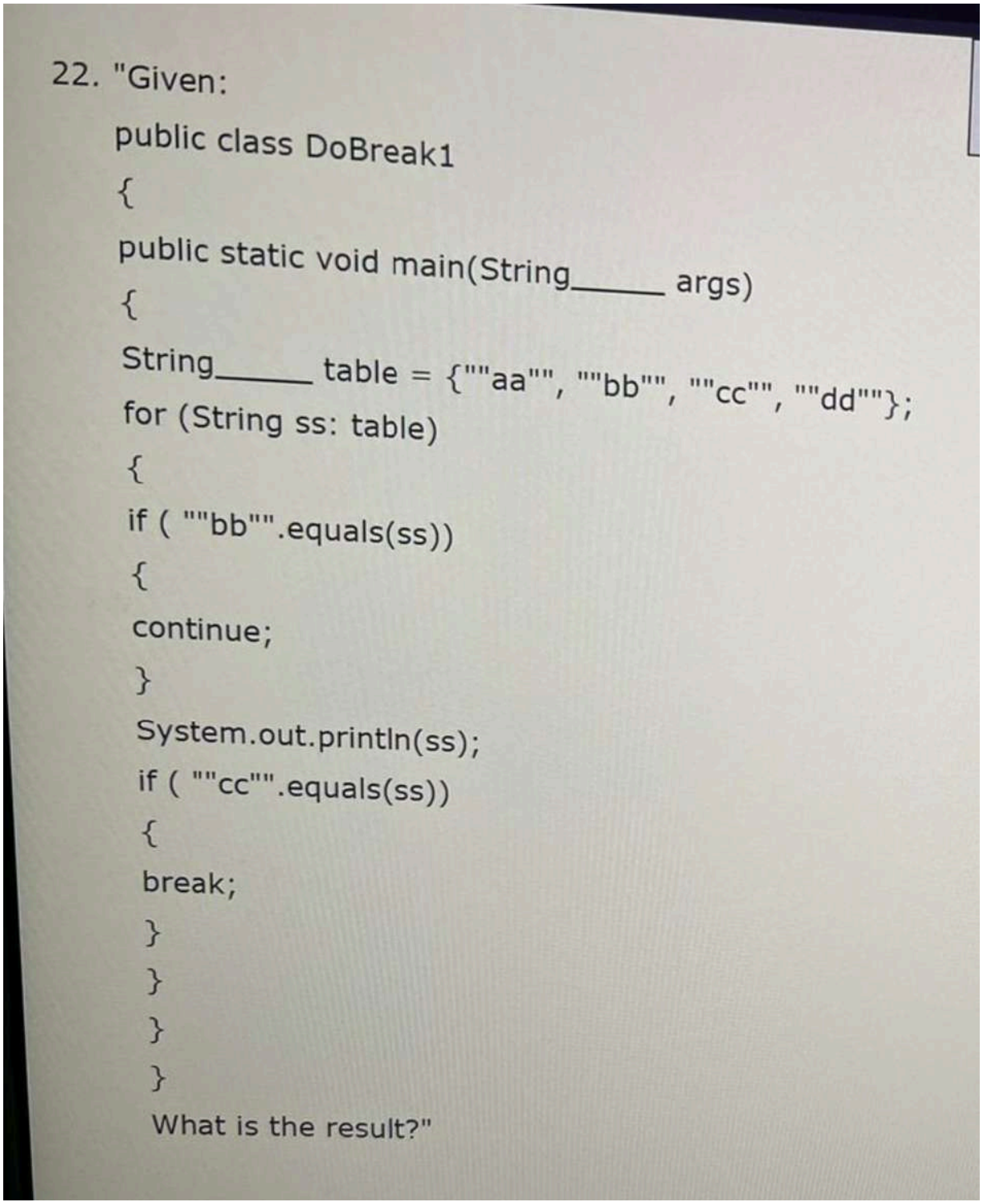
A. sb.append(“My dog” + name + ' is "'+ age); System.out.println(sb);

B. sb.insert("'My dog”'). append(name + ""is " + age); System.out.println(sb);

C. sb.insert("My dog "").insert(name).insert("' is "*).insert(age); System.out.println(sb);"

D. sb.append("My dog ").append (name).append(" is ").append (age); System.out.println(sb);

22. Given



```
22. "Given:
public class DoBreak1
{
public static void main(String_____ args)
{
String_____ table = {""aa"", ""bb"", ""cc"", ""dd""};
for (String ss: table)
{
if ( ""bb"".equals(ss))
{
continue;
}
System.out.println(ss);
if ( ""cc"".equals(ss))
{
break;
}
}
}
}
What is the result?"
```

A. aa cc
B. aa bb cc
C. cc dd
D. cc

23. Which three are valid types for switch?

A. int
B. float
C. Integer
D. String