Q.

```
interface Foo {}
class Alpha implements Foo {}
class Beta extends Alpha {}
class Delta extends Beta {
      public static void main( String[] args ) {
            Beta x = new Beta();
            16. //insert code here 16
      }
}
```

Which code, inserted at line 16, will cause a java.lang.ClassCastException?

A. Alpha a = x;
B. Foo f = (Delta)x;
C. Foo f = (Alpha)x;
D. Beta b = (Beta)(Alpha)x;

Answer: B

```
Foo(I)
   |
   |implements
   |
Alpa(C)
   |
   |extends
   |
Beta(C)  ======> x -> Foo f = (Delta)x; // invalid becoz the collecting type is of
Foo( which is not related)
   |
   |extends
   |
 Delta(C)
```

Q>

Given:

```
public class Batman {
   int squares = 81;
   public static void main(String[] args) {
     new Batman().go();
  }
   void go() {
    incr(++squares);
    System.out.println(squares);
   }
   void incr(int squares) { squares += 10; }
   }
```

What is the result?

A. 81
B. 82
C. 91
D. 92
E. Compilation fails.
F. An exception is thrown at runtime.

Answer: B

Given:

```
1. public class Pass {
```

```
2. public static void main(String [] args) {
3.          int x = 5;
4.          Pass p = new Pass();
5.          p.doStuff(x);
6.          System.out.print(" main x = " + x);
7. }
8.
9.    void doStuff(int x) {
10.         System.out.print(" doStuff x = " + x++);
11.   }
12.}
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. doStuff x = 6 main x = 6
D. doStuff x = 5 main x = 5
E. doStuff x = 5 main x = 6
F. doStuff x = 6 main x = 5

Answer: D

Q>
Given:

```
String[] elements = { "for", "tea", "too" };
String first = (elements.length > 0) ? elements[0] : null;
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. The variable first is set to null.
D. The variable first is set to elements[0].

Answer: D

Note:

```
int[] data = {10,20,30};
System.out.println(data.length);

String[] names = {"Navin","Haider","Nitin"};
System.out.println(names.length);
System.out.println(names[0].length());
```

Arrays => To find the length of the array we use length property or variable or field
String => To find the no of characters present in String we use length().

Q>Given:

```
10. public class SuperCalc {
11.         protected static int multiply(int a, int b) { return a * b;}
12. }
```

and:

```
20. public class SubCalc extends SuperCalc{
21.         public static int multiply(int a, int b) {
22.               int c = super.multiply(a, b);  // super => it always refers to object creation process.
23.               return c;
24.         }
25. }
```

and:

```
30. SubCalc sc = new SubCalc ();
31. System.out.println(sc.multiply(3,4));
32. System.out.println(SubCalc.multiply(2,2));
```

What is the result?
A. 12,4
B. The code runs with no output.
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 21.
E. Compilation fails because of an error in line 22.
F. Compilation fails because of an error in line 31.

Answer: E

Question

```
class Foo {
      public int a = 3; //instance variable
      public void addFive() { a += 5; System.out.print("f "); }
}
class Bar extends Foo {
      public int a = 8;//instance variable
      public void addFive() { this.a += 5; System.out.print("b " ); }//overrriding
}
```

Invoked with:

```
Foo f = new Bar();//loose coupling
f.addFive();//call will be decided by jvm based on runtime object becoz it is
overriding method
System.out.println(f.a);//since a is present in both parent and child compiler only
will bind based on the type
```

What is the result?
A. b 3
B. b 8
C. b 13
D. f 3
E. f 8
F. f 13
G. Compilation fails.
H. An exception is thrown at runtime.

Answer: A