```
public class Pass2 {
      public void main(String [] args) {
            int x = 6;
            Pass2 p = new Pass2();
            p.doStuff(x);
            System.out.print(" main x = " + x);
      }
      void doStuff(int x) {
            System.out.print(" doStuff x = " + x++);
      }
}
```
And the command-line invocations:
```
javac Pass2.java
java Pass2 5
```
What is the result?
A. Compilation fails.
B. An exception is thrown at runtime.
C. doStuff x = 6 main x = 6
D. doStuff x = 6 main x = 7
E. doStuff x = 7 main x = 6
F. doStuff x = 7 main x = 7

Answer:B

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

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

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

Answer: D

Given:

```
1. public class TestString1 {
2.     public static void main(String[] args) {
3.          String str = "420";
4.          str += 42;
5.          System.out.print(str);
6.    }
7. }
```

What is the output?

A. 42
B. 420
C. 462
D. 42042
E. Compilation fails.
F. An exception is thrown at runtime.

Answer: D

Given this code from Class B:

```
25. A a1 = new A();
26. A a2 = new A();
27. A a3 = new A();
28. System.out.println(A.getInstanceCount());
```

What is the result?

```
1. public class A{
2.
3. private int counter = 0;
4.
5. public static int getInstanceCount() {
6.    return counter;
7. }
8.
9. public A() {
10.   counter++;
11.}
12.
13.}
```

A. Compilation of class A fails.
B. Line 28 prints the value 3 to System.out.
C. Line 28 prints the value 1 to System.out.
D. A runtime error occurs when line 25 executes.
E. Compilation fails because of an error on line 28.

Answer: A

Given

```
1. public class Venus {
2.    public static void main(String[] args) {
3.          int[] x = { 1, 2, 3 };
4.          int y[] = { 4, 5, 6 };
5.          new Venus().go(x, y);
6.    }
7.
8.    void go(int[]... z) {
9.          for (int[] a : z)
10.               System.out.print(a[0]);
```

```
11.   }
12.}
```
What is the result?
A. 1
B. 12
C. 14
D. 123
E. Compilation fails.
F. An exception is thrown at runtime.

Answer: C

Given
```
public class Test{
  public static void main(String[] args){
            String[] arr = {"L","I","V","E"};
            int i=-2;

            if(i++==-1) arr[-(--i)]="F";
                else if (--i==-2)  arr[-++i] = "O";

            for(String s: arr) System.out.print(s);
  }
}
```
A. compilation error
B. An exception is thrown at runtime
C. LIVE
D. LIFE
E. LIVO
F. LOVE
G. LIOE

Answer: F

Given
```
 public class Test{
      Boolean b[] = new Boolean[2];
      public static void main(String... args){
            Test t= new Test();
            System.out.println(t.b[0] + ":" +t.b[1]);// null + ":" + null = >
null:null
      }
 }
```
A. NullpointerException
B. false:false
C. true:true
D. null:null
E. RunTimeException other than NullPointerException

Given
```
class Alien {
      String invade(short ships) { return "a few"; }
      String invade(short... ships) { return "many"; }
}
class Defender {
      public static void main(String [] args) {
```

```
            System.out.println(new Alien().invade(7));
      }
}
```

What is the result?
A. many
B. a few
C. Compilation fails
D. The output is not predictable
E. An exception is thrown at runtime

Answer: B

Inheritance

```
++++++++++
class Parent      //Parent class,Base Class, Super class
{

}
class Child extends Parent //Child class, Derived class, Sub class
{

}
```