

NO.1 Given:

```
package clothing;
public class Shirt {
    public static String getColor() {
        return "Green";
    }
}
```

Given the code fragment:

```
package clothing.pants;
// line n1
public class Jeans {
    public void matchShirt() {
        // line n2
        if(color.equals("Green")) {
            System.out.print("Fit");
        }
    }
    public static void main(String[] args) {
        Jeans trouser = new Jeans();
        trouser.matchShirt();
    }
}
```

Which two sets of actions, independently, enable the code fragment to print Fit?

- A.** At line n1 insert: import clothing.Shirt;At line n2 insert: String color = Shirt.getColor();
- B.** At line n1 insert: import clothing;At line n2 insert: String color = Shirt.getColor();
- C.** At line n1 insert: import static clothing.Shirt.getColor;At line n2 insert: String color = getColor();
- D.** At line n1 no changes required.At line n2 insert: String color = Shirt.getColor();
- E.** At line n1 insert: import Shirt;At line n2 insert: String color = Shirt.getColor();

Answer: A

NO.2 Which two are benefits of polymorphism? (Choose two.)

- A.** Faster code at runtime
- B.** More efficient code at runtime
- C.** More dynamic code at runtime
- D.** More flexible and reusable code
- E.** Code that is protected from extension by other classes

Answer: B D

NO.3 Given the code fragment:

```
public static void main(String[] args) {  
    int[][] arr = new int [2] [4];  
    arr[0] = new int []{1, 3, 5, 7};  
    arr[1] = new int []{1, 3};  
    for (int[] a : arr) {  
        for (int i : a) {  
            System.out.print(i+ " ");  
        }  
        System.out.println();  
    }  
}
```

What is the result?

A Compilation fails.

B

```
1 3  
1 3
```

C

```
1 3  
followed by an ArrayIndexOutOfBoundsException
```

D

```
1 3  
1 3 0 0
```

E

```
1 3 5 7  
1 3
```

A. Option A

- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: E

The screenshot shows an online Java IDE interface. The 'Your Code ...' section contains the following Java code:

```

1- public class MyClass {
2-     public static void main (String [] args) {
3-         int [][] arr = new int [2] [4];
4-         arr[0] = new int [] {1, 3, 5, 7};
5-         arr[1] = new int [] {1, 3};
6-         for (int [] a : arr) {
7-             for (int i : a) {
8-                 System.out.print(i+ " ");
9-             }
10-            System.out.println ();
11-        }
12-    }
13- }
14-

```

Below the code editor, the 'External Libraries ...' section has a button 'Add External Library (from Maven Repo)'. The 'CommandLine Arguments ...' section is empty. The 'Interactive mode : OFF' and 'Version: JDK 9.0.1' are displayed. The 'Stdin Inputs...' section is empty. Below these are buttons: 'Execute', 'Save', 'My Projects', 'Recent', 'Collaborate', and 'More Options ...'. The 'Result...' section shows the output of the code execution:

```

CPU Time: 0.13 sec(s), Memory: 30680 kilobyte(s)
compiled and executed in 0.705 sec(s)

1 3 5 7
1 3

```

NO.4 Given the code fragment:

```

public static void main(String[] args) {
    int ans;
    try {
        int num = 10;
        int div = 0;
        ans = num / div;
    } catch (ArithmeticException ae) {
        ans = 0; // line n1
    } catch (Exception e) {
        System.out.println("Invalid calculation");
    }
    System.out.println("Answer = " + ans); // line n2
}

```

What is the result?

- A. Answer = 0
- B. Invalid calculation

- C. Compilation fails only at line n1.
- D. Compilation fails only at line n2.
- E. Compilation fails at line n1 and line2.

Answer: C

```

1
2 public class Test {
3     public static void main(String[] args) {
4         int ans;
5         try {
6             int num = 10;
7             int div = 0;
8             ans = num / div;
9         } catch (ArithmeticException ae) {
10            ans = 0;
11        } catch (Exception e) {
12            System.out.println("Invalid calculation");
13        }
14        System.out.println("Answer = " + ans); //line n2
15    }
16 }
17

```

× variable ans might not have been initialized

NO.5 Given the code fragment:

```

3. public static void main(String[] args) {
4.     int x = 5;
5.     while (isAvailable(x)) {
6.         System.out.print(x);
7.
8.     }
9. }
10.
11. public static boolean isAvailable(int x) {
12.     return x-- > 0 ? true : false;
13. }

```

Which modification enables the code to print 54321?

- A. Replace line 6 with System.out. print (--x) ;
- B. At line 7, insert x --;
- C. Replace line 6 with --x; and, at line 7, insert System.out.print (x);
- D. Replace line 12 with return (x > 0) ? false: true;

Answer: B

NO.6 Given the code fragment:

```
6. char colorCode = 'y';
7. switch (colorCode) {
8.     case 'r':
9.         int color = 100;
10.        break;
11.    case 'b':
12.        color = 10;
13.        break;
14.    case 'y':
15.        color = 1;
16.        break;
17. }
18. System.out.println(color);
```

What is the result?

- A. It results in a compile time error at line 18.
- B. It results in a compile time error at line 9.
- C. It prints : 1
- D. It results in a compile time error at lines at lines 12 and 15.

Answer: A

NO.7 Given:

```
class Cart {
    Product p;
    double totalAmount;
}

class Product {
    String name;
    Double price;
}

public class Shop {
    public static void main(String[] args) {
        Cart c = new Cart();
        System.out.println(c.p + ":" + c.totalAmount);
    }
}
```


What is the result?

- A. null:null:0.0
- B. null:null
- C. <<HashCode>>:0.0
- D. null:0.0

Answer: D

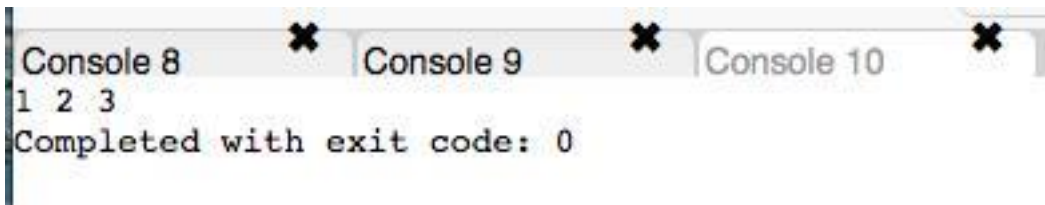
NO.8 Given the code fragment:

```
public static void main(String[] args) {  
    int[] arr = {1, 2, 3, 4};  
    int i = 0;  
    do {  
        System.out.print(arr[i] + " ");  
        i++;  
    } while (i < arr.length + 1);  
}
```

What is the result?

- A. 1 2 3 4 followed by an `ArrayIndexOutOfBoundsException`
- B. 1 2 3
- C. 1 2 3 4
- D. Compilation fails.

Answer: B



NO.9 Given:

```
public class Test {  
    // line n1  
}
```

Which two code fragments can be inserted at line n1?

- A. `String str = "Java";`
- B. `for(int iVal = 0; iVal <=5; iVal++){}`
- C. `Test() {}`
- D. `package p1;`
- E. `import java.io.*;`

Answer: A D

NO.10 Given the code fragment:

```
List<String> lst = Arrays.asList("EN", "FR", "CH", "JP");
Iterator<String> itr = lst.iterator();
while(itr.hasNext()) {
    String e = itr.next();
    if (e == "CH") {
        break;
    }
    System.out.print(e + " ");
}
```

What is the result?

- A. EN FR JP
- B. EN FR
- C. CH
- D. EN FR CH

Answer: B

```
16 public class Main {
17     public static void main(String[] args) {
18         List<String> lst = Arrays.asList("EN", "FR", "CH", "JP");
19         Iterator<String> itr = lst.iterator();
20         while(itr.hasNext()) {
21             String e = itr.next();
22             if(e == "CH") {
23                 break;
24             }
25             System.out.print(e+ " ");
26         }
27     }
28 }
```

Result

CPU Time: 0.28 sec(s), Memory: 35336 kilobyte(s)

EN FR

NO.11 What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A. Encapsulation
- B. Inheritance
- C. Abstraction
- D. Instantiation
- E. Polymorphism

Answer: A

Explanation

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

References:

NO.12 Given the definitions of the MyString class and the Test class:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8").msg);
    }
}
```

What is the result?

A

```
Hello Java SE 8  
Hello Java SE 8
```

B

```
Hello java.lang.StringBuilder@<<hashCode1>>  
Hello p1.MyString@<<hashCode2>>
```

C

```
Hello Java SE 8  
Hello p1.MyString@<<hashCode>>
```

D Compilation fails at the Test class

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: D**NO.13** Which two array initialization statements are valid? (Choose two.)

- A. `int array[] = new int[3] {1, 2, 3};`
- B. `int array[] = new int[3]; array[0] = 1;`
`array[1] = 2;`
`array[2] = 3;`
- C. `int array[3] = new int[] {1, 2, 3};`
- D. `int array[] = new int[3]; array = {1, 2, 3};`
- E. `int array[] = new int[] {1,2,3};`

Answer: B E**NO.14** Given:

```
public class MyClass {  
    public static void main(String[] args) {  
        String s = "Java SE 8 1";  
        int len = s.trim().length();  
        System.out.print(len);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 11
- C. 8
- D. 9
- E. 10

Answer: B

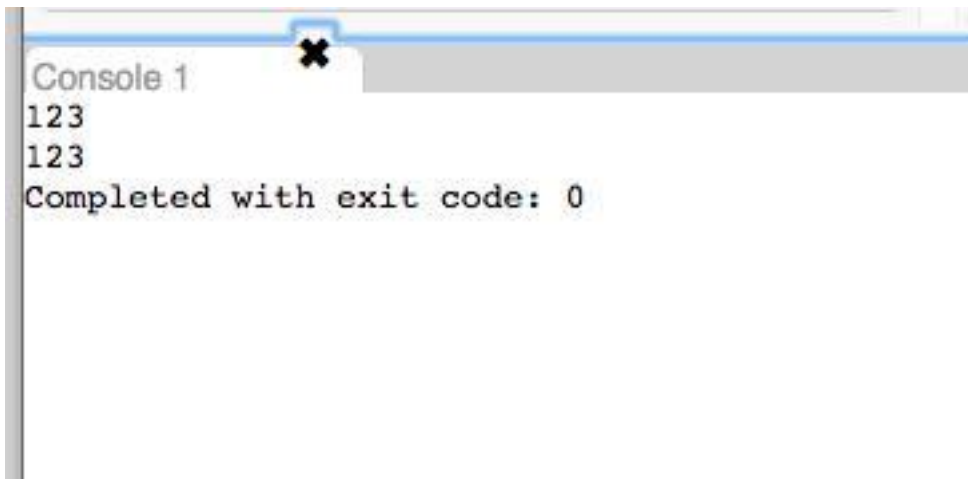
NO.15 Given the code fragment:

```
int array1[] = {1, 2, 3};  
int array2[] = new int [5];  
array2 = array1;  
for (int i : array2) {  
    System.out.print(i + " ");  
}  
System.out.println();  
int array3[] = new int[3];  
array3 = array2;  
for (int i : array3) {  
    System.out.print(i + " ");  
}
```

What is the result?

- A. 1 2 3 0 0 1 2 3 0 0
- B. An Exception is thrown at run time.
- C. 1 2 3 0 0 1 2 3
- D. 1 2 3 1 2 3

Answer: D



NO.16 Given:

```
public class Test {
    public static void main(String[] args) {
        int x = 1;
        int y = 1;
        if(x++ < ++y){
            System.out.print("Hello ");
        } else {
            System.out.print("Welcome ");
        }
        System.out.print("Log " + x + ":" + y);
    }
}
```

What is the result?

- A. Hello Log 2:2
- B. Welcome Log 1:2
- C. Welcome Log 2:1
- D. Hello Log 1:2

Answer: A

```
1 public class Main {
2     public static void main(String[] args) {
3         int x = 1;
4         int y = 1;
5         if (x++ < ++y) {
6             System.out.print("Hello ");
7         } else {
8             System.out.print("Welcome ");
9         }
10        System.out.print("Log " + x + ":" + y);
11    }
12 }
```

```
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
> javac -classpath ./run_dir/junit-4.12.jar:./run_dir/hamcrest-core-1.3.jar:./run_dir/json-simple-1.1.1.jar -d . Main.java
> java -classpath ./run_dir/junit-4.12.jar:./run_dir/hamcrest-core-1.3.jar:./run_dir/json-simple-1.1.1.jar Main
Hello Log 2:2
```

NO.17 Given the code fragment:

```
public static void main(String[] args) {  
    String str = " ";  
    str.trim();  
    System.out.println(str.equals("") + " " + str.isEmpty());  
}
```

What is the result?

- A. true true
- B. true false
- C. false false
- D. false true

Answer: C

NO.18 Given the code fragment:

```
public static void main(String[] args) {  
    String[] arr = {"A", "B", "C", "D"};  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
        if (arr[i].equals("C")) {  
            continue;  
        }  
        System.out.println("Work done");  
        break;  
    }  
}
```

What is the result?

- A. A B C Work done
- B. A B C D Work done
- C. A Work done
- D. Compilation fails

Answer: C

NO.19 Given the definitions of the MyString class and the Test class:

MyString.java:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8"));
    }
}
```

What is the result?

A

```
Hello Java SE 8
Hello Java SE 8
```

B

```
Hello java.lang.StringBuilder@<<hashCode1>>
Hello p1.MyString@<<hashCode2>>
```

C

```
Hello Java SE 8
Hello p1.MyString@<<hashCode>>
```

D Compilation fails at the Test class

A. Option A

B. Option B

C. Option C

D. Option D

Answer: C

NO.20 Which statement will empty the contents of a StringBuilder variable named sb?

A. sb.deleteAll();

B. sb.delete(0, sb.size());

C. sb. delete (0, sb. length ());

D. sb. removeAll ();

Answer: C