

**1z0-808.exam.84q**

Number: 1z0-808  
Passing Score: 800  
Time Limit: 120 min



**Website:** <https://vceplus.com>  
**VCE to PDF Converter:** <https://vceplus.com/vce-to-pdf/>  
**Facebook:** <https://www.facebook.com/VCE.For.All.VN/>  
**Twitter :** [https://twitter.com/VCE\\_Plus](https://twitter.com/VCE_Plus)

<https://vceplus.com/>

**1z0-808**

**Java SE 8 Programmer I**

## Exam A

### QUESTION 1

Given the code fragment:

```
1. class X {  
2.     public void printFileContent() {  
3.         /* code goes here */  
4.         throw new IOException();  
5.     }  
6. }  
7. public class Test {  
8.     public static void main(String[] args) {  
9.         X xobj = new X();  
10.        xobj.printFileContent();  
11.    }  
12. }
```

Which two modifications should you make so that the code compiles successfully? (Choose two.)

- ☐ A) Replace line 8 with `public static void main(String[] args) throws Exception {`
- ☐ B) Replace line 10 with:  

```
try {  
    xobj.printFileContent();  
}  
catch(Exception e) { }  
catch(IOException e) { }
```
- ☐ C) Replace line 2 with `public void printFileContent() throws IOException {`
- ☐ D) Replace line 4 with `throw IOException("Exception raised");`
- ☐ E) At line 11, insert `throw new IOException();`



<https://vceplus.com/>

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

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

## QUESTION 2

Given the following two classes:



```
public class Customer {
    ElectricAccount acct = new ElectricAccount();

    public void useElectricity(double kWh) {
        acct.addKWh(kWh);
    }
}

public class ElectricAccount {
    private double kWh;
    private double rate = 0.07;
    private double bill;

    //line n1
}
```

How should you write methods in the ElectricAccount class at line n1 so that the member variable bill is always equal to the value of the member variable kWh multiplied by the member variable rate?

Any amount of electricity used by a customer (represented by an instance of the customer class) must contribute to the customer's bill (represented by the member variable bill) through the method use Electricity method. An instance of the customer class should never be able to tamper with or decrease the value of the member variable bill.

- ☐ A) 

```
public void addKWh(double kWh) {  
    this.kWh += kWh;  
    this.bill = this.kWh*this.rate;  
}
```
- ☐ B) 

```
public void addKWh(double kWh) {  
    if (kWh > 0){  
        this.kWh += kWh;  
        this.bill = this.kWh * this.rate;  
    }  
}
```
- ☐ C) 

```
private void addKWh(double kWh) {  
    if (kWh > 0) {  
        this.kWh += kWh;  
        this.bill = this.kWh*this.rate;  
    }  
}
```
- ☐ D) 

```
public void addKWh(double kWh) {  
    if(kWh > 0) {  
        this.kWh += kWh;  
        setBill(this.kWh);  
    }  
}  
public void setBill(double kWh) {  
    bill = kWh*rate;  
}
```

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

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

### QUESTION 3

Given the code fragment:

```
public static void main(String[] args) {  
    StringBuilder sb = new StringBuilder(5);  
    String s = "";  
  
    if (sb.equals(s)) {  
        System.out.println("Match 1");  
    } else if (sb.toString().equals(s.toString())) {  
        System.out.println("Match 2");  
    } else {  
        System.out.println("No Match");  
    }  
}
```

What is the result?

- A. Match 1
- B. Match 2
- C. No Match
- D. A NullPointerException is thrown at runtime.

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

### QUESTION 4

Given:

```
interface Readable {  
    public void readBook();  
    public void setBookMark();  
}  
  
abstract class Book implements Readable { // line n1  
    public void readBook() { }  
    // line n2  
}  
  
class EBook extends Book { // line n3  
    public void readBook() { }  
    // line n4  
}
```

And given the code fragment:

```
Book book1 = new EBook ();  
Book1.readBook();
```



Which option enables the code to compile?

- ☐ A) Replace the code fragment at line n1 with:  
class Book implements Readable {
- ☐ B) At line n2 insert:  
public abstract void setBookMark();
- ☐ C) Replace the code fragment at line n3 with:  
abstract class EBook extends Book {
- ☐ D) At line n4 insert:  
public void setBookMark() { }

- A. Option A
- B. Option B

- C. Option C
- D. Option D

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 5

Given:

```
public static void main(String[] args) {  
    String ta = "A ";  
    ta = ta.concat("B ");  
    String tb = "C ";  
    ta = ta.concat(tb);  
    ta.replace('C', 'D');  
    ta = ta.concat(tb);  
    System.out.println(ta);  
}
```



What is the result?

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

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**



**QUESTION 6**

Given:

```
class CD {
    int r;
    CD(int r){
        this.r=r;
    }
}

class DVD extends CD {
    int c;
    DVD(int r, int c) {
        // line n1
    }
}
```

And given the code fragment:

```
DVD dvd = new DVD(10,20);
```



Which code fragment should you use at line n1 to instantiate the dvd object successfully?

- ☐ A) `super.r = r;`  
    `this.c = c;`
- ☐ B) `super(r);`  
    `this(c);`
- ☐ C) `super(r);`  
    `this.c = c;`
- ☐ D) `this.c = r;`  
    `super(c);`

- A. Option A
- B. Option B

- C. Option C
- D. Option D

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 7

Given the code fragment:

```
int a[] = {1, 2, 3, 4, 5};  
for (XXX) {  
    System.out.print(a[e]);  
}
```

Which option can replace xxx to enable the code to print 135?

```
int e = 0; e <= 4; e++  
int e = 0; e < 5; e += 2  
int e = 1; e <= 5; e += 1  
int e = 1; e < 5; e += 2
```

- A.
- B.
- C.
- D.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 8

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
- B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
- C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
- D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 9

Given the code fragment from three files:

SalesMan.java:

```
package sales;  
public class SalesMan { }
```

Product.java:

```
package sales.products;  
public class Product { }
```

Market.java:

```
1. package market;  
2. // insert code here  
3. public class USMarket {  
4.     SalesMan sm;  
5.     Product p;  
6. }
```



Which code fragment, when inserted at line 2, enables the code to compile?

- ☐ A) `import sales.*;`
- ☐ B) `import java.sales.products.*;`
- ☐ C) `import sales;`  
`import sales.products;`
- ☐ D) `import sales.*;`  
`import products.*;`
- ☐ E) `import sales.*;`  
`import sales.products.*;`



<https://vceplus.com/>

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

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 10**

Given the following class:

```
public class CheckingAccount {  
    public int amount;  
    public CheckingAccount(int amount){  
        this.amount = amount;  
    }  
    public int getAmount(){  
        return amount;  
    }  
    public void changeAmount(int x){  
        amount += x;  
    }  
}
```

And given the following main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount((int) (Math.random()*1000));  
    //line n1  
    System.out.println(acct.getAmount());  
}
```

Which three lines, when inserted independently at line n1, cause the program to print a 0 balance? (Choose three.)

- A. this.amount = 0;
- B. amount = 0;
- C. acct (0);
- D. acct.amount = 0;
- E. acct. getAmount () = 0;
- F. acct.changeAmount(0);
- G. acct.changeAmount(-acct.amount);
- H. acct.changeAmount(-acct.getAmount());

**Correct Answer:** ACD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 11

Given the code fragment:

```
String shirts[][] = new String[2][2];  
shirts[0][0] = "red";  
shirts[0][1] = "blue";  
shirts[1][0] = "small";  
shirts[1][1] = "medium";
```

Which code fragment prints red: blue: small: medium?



```
☐ A) for (int index = 1; index < 2; index++) {  
    for (int idx = 1; idx < 2; idx++) {  
        System.out.print(shirts[index][idx] + ":");  
    }  
}  
  
☐ B) for (int index = 0; index < 2; ++index) {  
    for (int idx = 0; idx < index; ++idx) {  
        System.out.print(shirts[index][idx] + ":");  
    }  
}  
  
☐ C) for (String c : colors) {  
    for (String s : sizes) {  
        System.out.println(s + ":");  
    }  
}  
  
☐ D) for (int index = 0; index < 2;) {  
    for (int idx = 0; idx < 2;) {  
        System.out.print(shirts[index][idx] + ":");  
        idx++;  
    }  
    index++;  
}
```

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

**Correct Answer:** D

**Section:** (none)

**Explanation****Explanation/Reference:****QUESTION 12**

Given the following main method:

```
public static void main(String[] args) {  
    int num = 5;  
    do {  
        System.out.print(num-- + " ");  
    } while (num == 0);  
}
```

What is the result?

- A. 5 4 3 2 1 0
- B. 5 4 3 2 1
- C. 4 2 1
- D. 5
- E. Nothing is printed



**Correct Answer:** D

**Section:** (none)

**Explanation****Explanation/Reference:****QUESTION 13**

Given the code fragment:

```
int x = 100;  
int a = x++;  
int b = ++x;  
int c = x++;  
int d = (a < b) ? (a < c) ? a : (b < c) ? b : c;  
System.out.println(d);
```



What is the result?

- A. 100
- B. 101
- C. 102
- D. 103
- E. Compilation fails

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 14

Given:



```
public class Test {  
    public static void main(String[] args) {  
        String[][] chs = new String[2][];  
        chs[0] = new String[2];  
        chs[1] = new String[5];  
        int i = 97;  
  
        for (int a = 0; a < chs.length; a++) {  
            for (int b = 0; b < chs[a].length; b++) {  
                chs[a][b] = "" + i;  
                i++;  
            }  
        }  
  
        for (String[] ca : chs) {  
            for (String c : ca) {  
                System.out.print(c + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

What is the result?

- A. 97 98  
99 100 null null null
- B. 97 98  
99 100 101 102 103C. Compilation fails.
- D. A NullPointerException is thrown at runtime.
- E. An ArrayIndexOutOfBoundsException is thrown at runtime.

**Correct Answer:** A

Section: (none)

Explanation

Explanation/Reference:

#### QUESTION 15

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString(){  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print `joe:true: 100.0`? (Choose two.)

- ☐ A) Replace line n2 with:  
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
- ☐ B) Replace line n2 with:  
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
- ☐ C) Replace line n1 with:  
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
- ☐ D) Replace line n1 with:  
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
- ☐ E) Replace line n1 with:  
this("Joe", true, 100);



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

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 16**

Given the code fragment:

```
public static void main(String[] args) {  
    List<String> names = new ArrayList<>();  
    names.add("Robb");  
    names.add("Bran");  
    names.add("Rick");  
    names.add("Bran");  
  
    if (names.remove("Bran")) {  
        names.remove("Jon");  
    }  
    System.out.println(names);  
}
```

What is the result?

- A. [Robb, Rick, Bran]
- B. [Robb, Rick]
- C. [Robb, Bran, Rick, Bran]
- D. An exception is thrown at runtime.



**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 17

Given:

```
class A {  
    public A() {  
        System.out.print("A ");  
    }  
}  
  
class B extends A {  
    public B() { //line n1  
        System.out.print("B ");  
    }  
}  
  
class C extends B {  
    public C() { //line n2  
        System.out.print("C ");  
    }  
    public static void main(String[] args) {  
        C c = new C();  
    }  
}
```

What is the result?

- A. C B A B.
- C
- C. A B C
- D. Compilation fails at line n1 and line n2

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 18**

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```



What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 19

Given the code fragment:

```
1. public class Test {  
2.     public static void main(String[] args) {  
3.         /* insert code here */  
4.         array[0]=10;  
5.         array[1]=20;  
6.         System.out.print (array[0]+": "+array[1]);  
7.     }  
8. }
```

Which code fragment, when inserted at line 3, enables the code to print 10:20?

`int[] array n= new int[2];` A.

`int[] array;  
array = int[2];`

`int array = new int[2];`

`int array [2];`

B.

C.

D.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 20

Which three statements describe the object-oriented features of the Java language? (Choose three.)



- A. Objects cannot be reused.
- B. A subclass can inherit from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain more than one class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Correct Answer:** BCF

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Reference: <http://www.javaworld.com/article/2075459/java-platform/java-101--object-oriented-language-basics--part-5--object-and-its-methods.html> (see the sub title, Object is root of all classes not all other objects)

#### QUESTION 21

Given the following code:

```
public static void main(String[] args){  
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};  
  
    System.out.println(planets.length);  
    System.out.println(planets[1].length());  
}
```

What is the output?

- A. 4  
4
- B. 3  
5
- C. 4  
7
- D. 5  
4
- E. 4  
5



<https://vceplus.com/>

F. 4  
21

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**



### QUESTION 22

You are developing a banking module. You have developed a class named ccMask that has a maskcc method.

Given the code fragment:

```
class CCMask {  
    public static String maskCC(String creditCard) {  
        String x = "XXXX-XXXX-XXXX-";  
        //line n1  
    }  
  
    public static void main(String[] args) {  
        System.out.println(maskCC("1234-5678-9101-1121"));  
    }  
}
```

You must ensure that the maskcc method returns a string that hides all digits of the credit card number except the four last digits (and the hyphens that separate each group of four digits).

Which two code fragments should you use at line n1, independently, to achieve this requirement? (Choose two.)

- ☐ A) 

```
StringBuilder sb = new StringBuilder(creditCard);  
sb.substring(15, 19);  
return x + sb;
```
- ☐ B) 

```
return x + creditCard.substring(15, 19);
```
- ☐ C) 

```
StringBuilder sb = new StringBuilder(x);  
sb.append(creditCard, 15, 19);  
return sb.toString();
```
- ☐ D) 

```
StringBuilder sb = new StringBuilder(creditCard);  
StringBuilder s = sb.insert(0, x);  
return s.toString();
```

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



**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 23

Given:

Acc.java:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}
```

Test.java:

```
package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```



Which statement is true?

- A. Both p and s are accessible by obj.
- B. Only s is accessible by obj.
- C. Both r and s are accessible by obj.
- D. p, r, and s are accessible by obj.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 24

Given:

Base.java:

```
class Base {  
    public void test() {  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test() {  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test() {  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        b1 = (Base) b3;  
        Base b4 = (DerivedA) b3;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

A. Base

- DerivedA
- B. Base  
DerivedB
- C. DerivedB  
DerivedB
- D. DerivedB  
DerivedA
- E. A ClassCastException is thrown at runtime.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 25

Given the code fragment:

```
public static void main(String[] args) {  
    ArrayList myList = new ArrayList();  
    String[] myArray;  
    try {  
        while (true) {  
            myList.add("My String");  
        }  
    }  
    catch (RuntimeException re) {  
        System.out.println("Caught a RuntimeException");  
    }  
    catch (Exception e) {  
        System.out.println("Caught an Exception");  
    }  
    System.out.println("Ready to use");  
}
```



What is the result?

- A. Execution terminates in the first catch statement, and caught a RuntimeException is printed to the console.
- B. Execution terminates in the second catch statement, and caught an Exception is printed to the console.
- C. A runtime error is thrown in the thread "main".

- D. Execution completes normally, and Ready to use is printed to the console.
- E. The code fails to compile because a throws keyword is required.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 26

Given:

```
System.out.println("5 + 2 = " + 3 + 4);  
System.out.println("5 + 2 = " + (3 + 4));
```

What is the result?

- ☐ A)  $5 + 2 = 34$   
 $5 + 2 = 34$
- ☐ B)  $5 + 2 + 3 + 4$   
 $5 + 2 = 7$
- ☐ C)  $7 = 7$   
 $7 + 7$
- ☐ D)  $5 + 2 = 34$   
 $5 + 2 = 7$



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

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 27**

Given the code fragments:





Person.java:

```
public class Person {
    String name;
    int age;

    public Person(String n, int a) {
        name = n;
        age = a;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {
    for (Person p : list) {
        if (predicate.test(p)) {
            System.out.println(p.name + " ");
        }
    }
}

public static void main(String[] args) {
    List<Person> iList = Arrays.asList(new Person("Hank", 45),
                                       new Person("Charlie", 40),
                                       new Person("Smith", 38));

    //line n1
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

A. `checkAge (iList, ( ) -> p. get Age ( ) > 40);`

```
checkAge(iList, Person p -> p.getAge( ) > 40);  
checkAge (iList, p -> p.getAge ( ) > 40);  
checkAge(iList, (Person p) -> { p.getAge() > 40; });
```

B.

C.

D.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 28**

Given:



```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

- ☐ A) `int sum is 30`  
`float sum is 30.0`
- ☐ B) `int sum is 30`  
`double sum is 30`
- ☐ C) `Integer sum is 30`  
`double sum is 30.0`
- ☐ D) `Integer sum is 30`  
`float sum is 30.0`

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

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 29

Given the code fragment:



```
String[] strs = new String[2];
int idx = 0;
for (String s : strs) {
    strs[idx].concat(" element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

What is the result?

- A. Element 0  
Element 1
- B. Null element 0  
Null element 1
- C. Null  
Null
- D. A NullPointerException is thrown at runtime.



**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 30

Given:

```
class Vehicle {
    int x;
    Vehicle(){
        this(10); // line n1
    }
    Vehicle(int x) {
        this.x = x;
    }
}

class Car extends Vehicle {
    int y;
    Car() {
        super();
        this(20); // line n2
    }
    Car(int y) {
        this.y = y;
    }
    public String toString() {
        return super.x + ":" + this.y;
    }
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();
System.out.println(y);
```

What is the result?

- A. 10:20
- B. 0:20

- C. Compilation fails at line n1
- D. Compilation fails at line n2

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 31

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 pl.MyString@<<hashCode2>>`
- ☐ C) `Hello Java SE 8`  
`Hello pl.MyString@<<hashCode>>`
- ☐ D) Compilation fails at the `Test` class.

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

**Correct Answer:** C  
**Section:** (none)  
**Explanation**



**Explanation/Reference:**

### QUESTION 32

Given the code fragment:

C. Line	2
D. Line	10
E. Line	10
F. Line	25
	26

**Correct Answer:** ADF

**Section:** (none)**Explanation****Explanation/Reference:**QUESTION

26  
 26



```
3. public static void main(String[] args) {  
4.     int iVar = 100;  
5.     float fVar = 100.100f;  
6.     double dVar = 123;  
7.     iVar = fVar;  
8.     fVar = iVar;  
9.     dVar = fVar;  
10.    fVar = dVar;  
11.    dVar = iVar;  
12.    iVar = dVar;  
13. }
```

Which three lines fail to compile?

- A. Line 7
- B. Line 8

Given:



MainTest.java:

```
public class MainTest {  
  
    public static void main(int[] args) {  
        System.out.println("int main " + args[0]);  
    }  
    public static void main(Object[] args) {  
        System.out.println("Object main " + args[0]);  
    }  
    public static void main(String[] args) {  
        System.out.println("String main " + args[0]);  
    }  
}
```

and commands:

```
javac MainTest.java  
java MainTest 1 2 3
```



What is the result?

- A. int main 1
- B. Object main 1
- C. String main 1
- D. Compilation fails
- E. An exception is thrown at runtime

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 34**

Given the code fragment:

```
int num[][] = new int[1][3];  
for (int i = 0; i < num.length; i++) {  
    for (int j = 0; j < num[i].length; j++) {  
        num[i][j] = 10;  
    }  
}
```

Which option represents the state of the num array after successful completion of the outer loop?

- ☐ A) num[0][0]=10  
num[0][1]=10  
num[0][2]=10
- ☐ B) num[0][0]=10  
num[1][0]=10  
num[2][0]=10
- ☐ C) num[0][0]=10  
num[0][1]=0  
num[0][2]=0
- ☐ D) num[0][0]=10  
num[0][1]=10  
num[0][2]=10  
num[0][3]=10  
num[1][0]=0  
num[1][1]=0  
num[1][2]=0  
num[1][3]=0





<https://vceplus.com/>

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

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 35**

Given the code fragment:



```
public class Person {
    String name;
    int age = 25;

    public Person(String name) {
        this();
        setName(name);
    }

    public Person(String name, int age) {
        Person(name);
        setAge(age);
    }

    //setter and getter methods go here

    public String show() {
        return name + " " + age + " " + number;
    }

    public static void main(String[] args) {
        Person p1 = new Person("Jesse");
        Person p2 = new Person("Walter", 52);
        System.out.println(p1.show());
        System.out.println(p2.show());
    }
}
```

What is the result?

Jesse 25

A. Walter 52

- B. Compilation fails only at line n1
- C. Compilation fails only at line n2
- D. Compilation fails at both line n1 and line n2

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 36

Given the following code for a Planet object:

```
public class Planet {  
    public String name;  
    public int moons;  
  
    public Planet(String name, int moons) {  
        this.name = name;  
        this.moons = moons;  
    }  
}
```

And the following main method:

```
public static void main(String[] args) {  
    Planet[] planets = {  
        new Planet("Mercury", 0),  
        new Planet("Venus", 0),  
        new Planet("Earth", 1),  
        new Planet("Mars", 2)  
    };  
  
    System.out.println(planets);  
    System.out.println(planets[2]);  
    System.out.println(planets[2].moons);  
}
```

What is the output?

- ☐ A) planets  
Earth  
1
- ☐ B) [LPlanets.Planet;@15db9742  
Earth  
1
- ☐ C) [LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
1
- ☐ D) [LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
[LPlanets.Moon;@7852e922
- ☐ E) [LPlanets.Planet;@15db9742  
Venus  
0



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

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 37**

You are asked to develop a program for a shopping application, and you are given the following information:

- The application must contain the classes Toy, EduToy, and ConsToy. The Toy class is the superclass of the other two classes.
- The int calculatePrice (Toy t) method calculates the price of a toy. ▪ The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

- ☐ A) 

```
public abstract class Toy{  
    public abstract int calculatePrice(Toy t);  
    public void printToy(Toy t) { /* code goes here */ }  
}
```
- ☐ B) 

```
public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public void printToy(Toy t) ;  
}
```
- ☐ C) 

```
public abstract class Toy {  
    public int calculatePrice(Toy t);  
    public final void printToy(Toy t){ /* code goes here */ }  
}
```
- ☐ D) 

```
public abstract class Toy {  
    public abstract int calculatePrice(Toy t) { /* code goes here */ }  
    public abstract void printToy(Toy t) { /* code goes here */ }  
}
```

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

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 38

Given the following code:

```
int[] intArr = {15, 30, 45, 60, 75};  
intArr[2] = intArr[4];  
intArr[4] = 90;
```

What are the values of each element in intArr after this code has executed?



- A. 15, 60, 45, 90, 75
- B. 15, 90, 45, 90, 75
- C. 15, 30, 75, 60, 90
- D. 15, 30, 90, 60, 90
- E. 15, 4, 45, 60, 90

**Correct Answer:** C

**Section:** (none)

**Explanation**

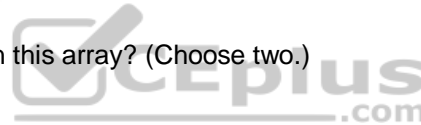
**Explanation/Reference:**

#### QUESTION 39

Given the following array:

```
int[] intArr = {8, 16, 32, 64, 128};
```

Which two code fragments, independently, print each element in this array? (Choose two.)



- ☐ A) `for (int i : intArr) {  
    System.out.print(intArr[i] + " ");  
}`
- ☐ B) `for (int i : intArr) {  
    System.out.print(i + " ");  
}`
- ☐ C) `for (int i=0 : intArr) {  
    System.out.print(intArr[i] + " ");  
    i++;  
}`
- ☐ D) `for (int i=0; i < intArr.length; i++) {  
    System.out.print(i + " ");  
}`
- ☐ E) `for (int i=0; i < intArr.length; i++) {  
    System.out.print(intArr[i] + " ");  
}`
- ☐ F) `for (int i; i < intArr.length; i++) {  
    System.out.print(intArr[i] + " ");  
}`

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

**Correct Answer:** BE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 40**

Given the content of three files:

A.java:

```
public class A {  
    public void a() {}  
    int a;  
}
```

B.java:

```
public class B {  
    private int doStuff() {  
        private int x = 100;  
        return x++;  
    }  
}
```

C.java:

```
import java.io.*;  
package p1;  
class A {  
    public void main(String fileName) throws IOException { }  
}
```

Which statement is true?

- A. Only the A.Java file compiles successfully.
- B. Only the B.java file compiles successfully.
- C. Only the C.java file compiles successfully.
- D. The A.Java and B.java files compile successfully.

- E. The B.java and C.java files compile successfully.
- F. The A.java and C.java files compile successfully.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 41

Given the code fragment:

```
int[] array = {1, 2, 3, 4, 5};
```

And given the requirements:

1. Process all the elements of the array in the order of entry.
2. Process all the elements of the array in the reverse order of entry.
3. Process alternating elements of the array in the order of entry.

Which two statements are true? (Choose two.)

- A. Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.
- B. Requirements 1, 2, and 3 can be implemented by using the standard for loop.
- C. Requirements 2 and 3 CANNOT be implemented by using the standard for loop.
- D. Requirement 1 can be implemented by using the enhanced for loop.
- E. Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.

**Correct Answer:** DE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 42

Given:

```
public class TestScope {  
    public static void main(String[] args) {  
        int var1 = 200;  
        System.out.print(doCalc(var1));  
        System.out.print(" "+var1);  
    }  
    static int doCalc(int var1){  
        var1 = var1 * 2;  
        return var1;  
    }  
}
```

What is the result?

- A. 400 200
- B. 200 200
- C. 400 400
- D. Compilation fails.



**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 43

Given:

```
public class Product {  
    int id;  
    String name;  
    public Product(int id, String name) {  
        this.id = id;  
        this.name = name;  
    }  
}
```

And given the code fragment:

```
4. Product p1 = new Product(101, "Pen");  
5. Product p2 = new Product(101, "Pen");  
6. Product p3 = p1;  
7. boolean ans1 = p1 == p2;  
8. boolean ans2 = p1.name.equals(p2.name);  
9. System.out.print(ans1 + ":" + ans2);
```

What is the result?

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

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 44

Given the following classes:

```
public class Employee {  
    public int salary;  
}  
  
public class Manager extends Employee {  
    public int budget;  
}  
  
public class Director extends Manager {  
    public int stockOptions;  
}
```

And given the following main method:

```
public static void main(String[] args) {  
    Employee employee = new Employee();  
    Manager manager = new Manager();  
    Director director = new Director();  
    //line n1  
}
```

Which two options fail to compile when placed at line n1 of the main method? (Choose two.)

- A. employee.salary = 50\_000;
- B. director.salary = 80\_000;
- C. employee.budget = 200\_000;
- D. manager.budget = 1\_000\_000;
- E. manager.stockOption = 500;
- F. director.stockOptions = 1\_000;

**Correct Answer:** CE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 45**

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start();  
        c.init();  
    }  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. Initialized  
Started  
Initialized
- C. Initialized  
Started
- D. Compilation fails.

**Correct Answer: D**



Section: (none)

Explanation

Explanation/Reference:

#### QUESTION 46

Given the code fragment:

```
public static void main(String[] args) {  
    try {  
        int num = 10;  
        int div = 0;  
        int 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 only at line n1 and line2.

**Correct Answer:** E

Section: (none)

Explanation

Explanation/Reference:

#### QUESTION 47

Given:

```
public class MyField {  
    int x;  
    int y;  
    public void doStuff(int x, int y) {  
        this.x = x;  
        y = this.y;  
    }  
    public void display () {  
        System.out.print(x + " " + y + " : ");  
    }  
    public static void main(String[] args) {  
        MyField m1 = new MyField();  
        m1.x = 100;  
        m1.y = 200;  
        MyField m2 = new MyField();  
        m2.doStuff(m1.x, m1.y);  
        m1.display();  
        m2.display();  
    }  
}
```

What is the result?

- A. 100 0 : 100 200:
- B. 100 0 : 100 0 :
- C. 100 200 : 100 200 :
- D. 100 200 : 100 0 :

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 48**

Given:

```
public class Vowel {  
    private char var;  
    public static void main(String[] args) {  
        char var1 = 'a';  
        char var2 = var1;  
        var2 = 'e';  
  
        Vowel obj1 = new Vowel ();  
        Vowel obj2 = obj1;  
        obj1.var = 'i';  
        obj2.var = 'o';  
  
        System.out.println(var1 + ", " + var2);  
        System.out.print(obj1.var + ", " + obj2.var);  
    }  
}
```

- A. a, ei,  
o
- B. a, e  
o, o
- C. e, ei,  
o

D. e,  
eo, o

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 49

Given the code fragment:

```
if (aVar++ < 10) {  
    System.out.println(aVar + " Hello World!");  
} else {  
    System.out.println(aVar + " Hello Universe!");  
}
```

What is the result if the integer aVar is 9?

- A. Compilation fails.
- B. 10 Hello Universe!
- C. 10 Hello World!
- D. 9 Hello World!

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 50

Given:

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

What is the result?

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

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 51

Given:

```
public class Test {  
    public static void main(String[] args) {  
        boolean a = new Boolean(Boolean.valueOf (args[0]));  
        boolean b = new Boolean(args[1]);  
        System.out.println(a + " " + b);  
    }  
}
```

And given the commands:

```
javac Test.java
java Test TRUE null
```

What is the result?

- A. TRUE null
- B. true false
- C. false false
- D. true true
- E. A `ClassCastException` is thrown at runtime.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**



## QUESTION 52

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.

1 3

1 3

1 3

followed by an `ArrayIndexOutOfBoundsException`

1 3

1 3 0 0

1 3 5 7

1 3

B. C.

D.

E.



**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Your 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-

```

External Libraries ...

CommandLine Arguments ...

Interactive mode : ☐ OFF Version: JDK 9.0.1

Stdin Inputs...

Result...

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

```

1 3 5 7
1 3

```

### QUESTION 53

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 ();

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 54

Given:

```
String stuff = "TV";  
String res = null;
```

```
if (stuff.equals ("TV")) {  
    res = "Walter";  
} else if (stuff.equals ("Movie) ) {  
    res= "White";  
} else {  
    res= "No Result";  
}
```



Which code fragment can replace the if block?



<https://vceplus.com/> A.

```
stuff.equals ("TV") ? res= "Walter" : stuff.equals ("Movie") ?  
res = "White" : res = "No Result";  
res = stuff.equals ("TV") ? "Walter" else stuff.equals  
("Movie")? "White" : "No Result";  
res = stuff.equals ("TV") ? stuff.equals ("Movie")? "Walter" :  
"White" : "No Result";  
res = stuff.equals ("TV")? "Walter" : stuff.equals ("Movie")?  
"White" : "No Result";
```

B. C.

D.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 55

Given:

```
class Patient {  
    String name;  
    public Patient (String name) {  
        this.name = name;  
    }  
}
```

And the code fragment:

```
8. public class Test {  
9.     public static void main (String [] args) {  
10.         List ps = new ArrayList ();  
11.         Patient p2 = new Patient ("Mike");  
12.         ps.add(p2);  
13.  
14.         // insert code here  
15.  
16.         if (f >= 0) {  
17.             System.out.print ("Mike Found");  
18.         }  
19.     }  
20. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

```
int f = ps.indexOf (p2);  
int f = ps.indexOf (Patient ("Mike") );  
int f = ps.indexOf (new Patient "Mike" );  
  
Patient p = new Patient ("Mike");  
Int f = ps.indexOf (p)
```

- A.
- B.
- C.
- D.

**Correct Answer:** A

**Section:** (none)

**Explanation**



**Explanation/Reference:**

#### QUESTION 56

Which statement is true about the switch statement?

- A. It must contain the default section.
- B. The break statement, at the end of each case block, is mandatory.
- C. Its case label literals can be changed at runtime.
- D. Its expression must evaluate to a single value.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Reference: <http://www.dummies.com/programming/java/switch-statements-in-java/>

#### QUESTION 57

Given:

```
class Animal {
    String type = "Canine";
    int maxSpeed = 60;

    Animal () {}

    Animal (String type, int maxSpeed) {
        this.type = type;
        this.maxSpeed = maxSpeed;
    }
}

class WildAnimal extends Animal {
    String bounds;

    WildAnimal (String bounds) {
        //line n1
    }

    WildAnimal (String type, int maxSpeed, String bounds) {
        //line n2
    }
}
```

And given the code fragment:

```
7. WildAnimal wolf = new WildAnimal ("Long");
8. WildAnimal tiger = new WildAnimal ("Feline", 80, "Short");
9. System.out.println (wolf.type + " " + wolf.maxSpeed + " " +
wolf.bounds);
10. Sytem.out.println (tiger.type + " " + tiger.maxSpeed + " " +
tiger.bounds);
```

Which two modifications enable the code to print the following output? (Choose two.)

```
Canine 60 Long
Feline 80 Short A. .
```

Replace line n1 with:

```
super ();
this.bounds = bounds;
```



B. Replace line n1 with:

```
this.bounds = bounds;
super ();
```

C. Replace line n2 with:

```
super (type, maxSpeed);
this (bounds);
```

D. Replace line n1 with:

```
this ("Canine", 60);
this.bounds = bounds
```

E. Replace line n2 with:

```
super (type, maxSpeed);  
this.bounds = bounds;
```

**Correct Answer:** AE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 58

Given the code fragment:

```
public static void main (String [] args) {  
    String names [] = ("Thomas", "Peter", "Joseph");  
    String pwd [] = new String [3];  
    int idx = 0;  
    try {  
        for (String n: names) {  
            pwd [idx] = n.substring (2, 6);  
            idx++;  
        }  
    }  
    catch (Exception e) {  
        System.out.println ("Invalid Name");  
    }  
    for (String p: pwd) {  
        System.out.println (p);  
    }  
}
```

What is the result?

A. Invalid Name

Invalid Name B. C.

omas

Invalid Name

omas

null

null

omas

ter

seph

D.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 59**

Given the code fragment:





```
class Employee {  
    private String name;  
    private int age;  
    private int salary;  
  
    public Employee (String name, int age) {  
        setName (name)  
        setAge (age)  
        setSalary (2000);  
    }  
    public Employee (String name, int age, int salary) {  
        setSalary (salary);  
        this (name, age);  
    }  
    //getter and setter methods for attributes go here  
    public void printDetails () {  
        System.out.println (name + " : " + age + " : " + salary);  
    }  
}
```

Test.java

```
class Test {  
    public static void main (String [] args {  
        Employee e1 = new Employee ();  
        Employee e2 = new Employee ("Jack, 50);  
        Employee e3 = new Employee ("Chloe", 40, 5000);  
        e1.printDetails ();  
        e2.printDetails ();  
        e3.printDetails ();  
    }  
}
```

Which is the result?

A. Compilation fails in the Employee class.

    null : 0 : 0                      B. C.

    Jack : 50 : 0

    Chloe : 40 : 5000

    null : 0 : 0

    Jack : 50 : 2000

    Chloe : 40 : 5000

D. Compilation fails in the Test class.

E. Both the Employee class and the test class fail to compile.

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 60**

Given the code fragments:



A. java:

```
package p1;  
public class A {  
}
```

B. java:

```
package p1.p2;  
//line n1  
public class B {  
    public void doStuff () {  
        A b = new A ();  
    }  
}
```

C. java

```
package p3;  
//line n2  
public class C {  
    public static void main (String [] args) {  
        A 01 = new A ();  
        B 02 = new B ();  
    }  
}
```



Which modification enables the code to compile?

Replace line n1 with:

```
import p1.*;
```

Replace line n2 with:

```
import p1. p2.*;
```

Replace line n1 with:

```
import p1. A;
```

Replace line n2 with:

```
import p1.*;
```

Replace line n1 with:

```
import p1. A;
```

Replace line n2 with:

```
import p1. A;
```

```
import p1. p2.B ;
```

Replace line n1 with:

```
import p1;
```

Replace line n2 with:

```
import p1;|
```

```
import p1. p2;
```

A.

B. C.

D.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 61**

Given:



```
class A {
    public void test () {
        System.out.println ("A");
    }
}
class B extends A {
    public void test () {
        System.out.println ("B");
    }
}
public class C extends A {
    public void test () {
        System.out.println ("C");
    }

    public static void main (String [] args) {
        A b1 = new A ();
        A b2 = new C ();
        b1 = (A) b2;           //line n1
        A b3 = (B) b2;         //line n2
        b1.test ();
        b3.test ();
    }
}
```

What is the result?

- A. A  
B
- B. A  
C
- C. C  
C
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 62**

Given:



```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

```
int sum is 30  
float sum is 30.0  
  
int sum is 30  
double sum is 30.0  
  
integer sum is 30  
double sum is 30.0
```

A.



B. C.

```
integer sum is 30  
float sum is 30.0
```

D.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 63

Given the code fragment:

```
4. class X {  
5.     public void printFileContent () {  
6.         /* code goes here */  
7.         throw new IOException ();  
8.     }  
9. }  
10. public class Test {  
11.     public static void main (String [] args) {  
12.         X xobj = new X ();  
13.         xobj.printFileContent ();  
14.     }  
15. }
```

Which two modifications should you make so that the code compiles successfully?

- A. At line 14, insert `throw new IOException ();`
- B. Replace line 5 with `public void printFileContent () throws IOException {`
- C. Replace line 11 with `public static void main (String [] args) throws Exception {`
- D. Replace line 13 with:

```
try {  
    xobj.printFileContent ();  
}  
catch (Exception e) {}  
catch (IOException e) {}
```
- E. Replace line 7 with `throw IOException ("Exception raised");`

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

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 64

You are asked to create a method that accepts an array of integers and returns the highest value from that array.

Given the code fragment:

```
class Test {  
    public static void main (String [] args) {  
        int numbers [] = {12, 13, 42, 32, 15, 156, 23, 51, 12};  
        int max = findMax (numbers);  
    }  
    /*line n1 */ {  
        int max = 0;  
        /* code goes here*/  
        return max;  
    }  
}
```

Which method signature do you use at line n1?

- A. public int findMax (int [] numbers)
- B. static int[] findMax (int max)
- C. static int findMax (int [] numbers)
- D. final int findMax (int [] )

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 65

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A public class must have a main method.
- B. A class can have only one private constructor.
- C. A method can have the same name as a field.
- D. A class can have overloaded static methods.
- E. The methods are mandatory components of a class.

F. The fields need not be initialized before use.

**Correct Answer:** ACE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 66

Given the code fragment:

```
Public static void main (String [] args) {  
    System.out.println ("Result A " + 0 + 1);  
    System.out.println ("Result B " + (1) + (2) );  
}
```



What is the result?

- A. Result A 1  
Result B 3
- B. Result A 01  
Result B 3
- C. Result A 01  
Result B 12
- D. Result A 1  
Result B 12

- A. Option A
- B. Option B
- C. Option C

D. Option D

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 67

Given:

```
public class App {  
    int count;  
    public static void displayMsg () {  
        count++; // line n1  
        System.out.println ("Welcome "+"Visit Count: "+count); // line n2  
    }  
    public static void main (String [] args) {  
        App.displayMsg (); // line n3  
        App.displayMsg (); // line n4  
    }  
}
```

What is the result?

- A. Compilation fails at line n3 and line n4.
- B. Compilation fails at line n1 and line n2.
- C. Welcome Visit Count:1 Welcome Visit Count: 1
- D. Welcome Visit Count:1  
Welcome Visit Count: 2

**Correct Answer:** B

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 68**

Given the code fragment:



```
public class Person {
    String name;
    int age = 25;

    public Person (String name) {
        this (); // //line n1
        setName(name);
    }
    public Person (String name, int age) {
        Person (name); //line n2
        setAge (age);
    }
    //setter and getter methods go here

    public String show () {
        return name + " " + age;
    }
    public static void main (String [] args) {
        Person p1 = new Person ("Jesse");
        Person p2 = new Person ("Walter", 52);
        System.out.println (p1.show () );
        System.out.println (p2.show () );
    }
}
```



<https://vceplus.com/>

What is the result?

- A. Compilation fails at both line n1 and line n2.
- B. Compilation fails only at line n2.
- C. Compilation fails only at line n1.
- D. Jesse 25  
Walter 52

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 69

Given the code fragment:





```
public class Test {  
  
    static int count = 0  
    int i = 0;  
  
    public void changeCount () {  
        while (i<5) {  
            i++;  
            count++;  
        }  
    }  
  
    public static void main (String [] args) {  
        Test check1 = new Test ();  
        Test check2 = new Test ();  
        check1.changeCount ();  
        check2.changeCount ();  
        System.out. print (check1.count + " : " + check2.count);  
    }  
}
```

What is the result?

- A. 5 : 5
- B. 10 : 10
- C. 5 : 10
- D. Compilation fails.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Reference:



Version - JDK 1.8.0\_66

Your Code ...

```
1 public class Test {  
2  
3     static int count = 0 ;  
4     int i = 0;  
5  
6     public void changecount () {  
7         while (i<5) {  
8             i++;  
9             count++;  
10        }  
11    }  
12    public static void main (String [ ] args) {  
13        Test check1 = new Test () ;  
14        Test check2 = new Test () ;  
15        check1.changecount () ;  
16        check2.changecount () ;  
17        System.out. print (check1.count + " : " + check2.count) ;  
18    }  
19 }  
20
```

External Libraries ... [Add External Library \(from Maven Repo\)](#)

cs1.keyboard

Input Arguments (args of Main Method)...

Interactive mode : ☐ OFF

Stdin Inputs...

Execute

Save

My Projects

Recent

Collaborate

Others ▾

Goto Another Language/DB ▾

Result...

compiled and executed in 1.357 second(s)

10 : 10

**QUESTION 70**

Given the code fragment:

```
public static void main (String [] args) {  
    ArrayList<Integer> points = new ArrayList<> ();  
    points.add (1);  
    points.add (2);  
    points.add (3);  
    points.add (4);  
    points.add (null);  
    points.remove (2);  
    points.remove (null);  
    System.out.println(points);  
}
```



What is the result?

- A. A NullPointerException is thrown at runtime.
- B. [1, 2, 4]
- C. [1, 2, 4, null ]
- D. [1, 3, 4, null ]
- E. [1, 3, 4 ]
- F. Compilation fails.

**Correct Answer:** F

**Section:** (none)

**Explanation**

**Explanation/Reference:**


Explanation:

Version - JDK 1.8.0\_66

Your Code ...

```
1 public static void main (String [] args) {  
2     ArrayList<Integer> points = new ArrayList<> ();  
3     points.add (1) ;  
4     points.add (2) ;  
5     points.add (3) ;  
6     points.add (4) ;  
7     points.add (null) ;  
8     points.remove (null) ;  
9     System.out.println (points) ;  
10 }
```

External Libraries ...

 Add External Library (from Maven Repo)

cs1.keyboard

Input Arguments (args of Main Method)...

Interactive mode : ☐ OFF

Stdin Inputs...

Execute

Save

My Projects

Recent

Collaborate

Others ▾

Goto Another Language/DB ▾

Result...

compiled and executed in 0 second(s)

No "public class" found to execute

QUESTION 71 Given:

```
class Test {  
    public static void main (String [] args) {  
        int numbers [ ];  
        numbers = new int [2];  
        numbers [0] = 10;  
        numbers [1] = 20;  
  
        numbers = new int [4];  
        numbers [2] = 30;  
        numbers [3] = 40;  
        for (int x : numbers) {  
            System.out.print (" " + x) ;  
        }  
    }  
}
```



What is the result?

- A. 10 20 30 40
- B. 0 0 30 40
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

## QUESTION 72

Which code fragment causes a compilation error?

```
A.float flt = 100F;
B.float flt = (float) 1_11.00;
C.float flt = 100;
D.double y1 = 203.22;
   float flt = y1;
E.int y2 = 100;
   float flt = (float) y2;
```

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

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 73

Given the code fragment:

```
public static void main (String [ ] args) {
    int [] stack = {10,20,30};
    int size = 3;
    inti dx = 0;
    /*line n1 */
    System.out.print ("The Top element: " + stack [idx] );
}
```

Which code fragment, inserted at line n1, prints The Top element: 30?

- A. 

```
do {  
    idx++;  
} while (idx >= size);
```
- B. 

```
while (idx < size) {  
    idx++;  
}
```
- C. 

```
do {  
    idx++;  
} while (idx < size - 1);
```
- D. 

```
do {  
    idx++;  
} while (idx <= size);
```
- E. 

```
while (idx <= size - 1) {  
    idx++  
}
```



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

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 74**



Given the code fragment:

```
public static void main (String [] args) {  
    String myStr = "Hello World";  
    myStr.trim ()  
    int i1 = myStr.indexOf (" ");  
    System.out.println (i1);  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. -1
- C. 5
- D. 0

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 75**

Given:



```
class Equal {
    public static void main (String [] args) {
        String str1 = "Java";
        String [] str2 = { "J", "a", "v", "a"};
        String str3 = "";
        for (String str : str2) {
            str3 = str3+str;
        }
        boolean b1 = (str1== str3);
        boolean b2 = (str1.equals (str3));
        System.out.print (b1+",    "+b2);
    }
}
```

What is the result?

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

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 76

Which two statements are true? (Choose two.)

- A. Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.

E. Error is a Throwable.

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 77

Given the code fragment:

```
public static void main (String[ ] args) {  
    int data [] = {2010, 2013, 2014, 2015, 2014};  
    int key = 2014;  
    int count = 0;  
    for (int e: data) {  
        if (e! = key) {  
            continue;  
            count++;  
        }  
    }  
    System.out.print (count + "Found");  
}
```



What is the result?

- A. Compilation fails.
- B. 0 Found
- C. 1 Found
- D. 3 Found

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 78**

Given the code fragment:

```
LocalDate Time dt= LocalDateTime.of (2014, 7, 31, 1, 1);  
dt.plusDays (30);  
dt. plusMonths (1);  
System.out.print (dt format (DateTimeFormatter. ISO_DATE) );
```

What is the result?

- A. An exception is thrown at runtime.
- B. 07-31-2014
- C. 2014-07-31
- D. 2014-09-30



**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 79**

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 = 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 clothing;`

At line n2 insert: `String color = Shirt.getColor();`

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 80

Given the code fragments:

```
class Student {  
    String name;  
    int age;  
}
```

And,



```
4. public class Test {  
5.     public static void main(String[] args) {  
6.         Student s1 = new Student();  
7.         Student s2 = new Student();  
8.         Student s3 = new Student();  
9.         s1 = s3;  
10.        s3 = s2;  
11.        s2 = null;  
12.    }  
13. }
```

Which statement is true?

- A. After line 11, three objects are eligible for garbage collection.
- B. After line 11, two objects are eligible for garbage collection.
- C. After line 11, one object is eligible for garbage collection.
- D. After line 11, none of the objects are eligible for garbage collection.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 81

Given:

```
public class App {  
    public static void main(String[] args) {  
        int i = 10;  
        int j = 20;  
        int k = j += i / 5;  
        System.out.print(i + " : " + j + " : " + k);  
    }  
}
```

What is the result?

- A. 10 : 30 : 6
- B. 10 : 22 : 22
- C. 10 : 22 : 20
- D. 10 : 22 : 6

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation



Your Code ...

```
1 public class App {  
2     public static void main (String[] args) {  
3         int i = 10;  
4         int j = 20;  
5         int k = j += i / 5;  
6         System.out.print (i + " : " + j + " : " + k);  
7     }  
8 }  
9
```

External Libraries ... [Add External Library \(from Maven Repo\)](#)

CommandLine Arguments ...

Interactive mode : ☐ OFF

Version:

JDK 9.0.1

Stdin Inputs...

[Execute](#) [Save](#) [My Projects](#) [Recent](#) [Collaborate](#) [More Options](#)

Result...

CPU Time: 0.20 sec(s), Memory: 32080 kilobyte(s)

compiled and executed in 1.229 sec(s)

10 : 22 : 22

QUESTION 82

Given:

```
interface Downloadable {  
    public void download();  
}  
  
interface Readable extends Downloadable {           // line n1  
    public void readBook();  
}  
  
abstract class Book implements Readable {          // line n2  
    public void readBook() {  
        System.out.println("Read Book");  
    }  
}  
  
class EBook extends Book {                          // line n3  
    public void readBook() {  
        System.out.println("Read E-Book");  
    }  
}
```

And given the code fragment:

```
Book book1 = new EBook();  
book1.readBook();
```

What is the result?

- A. Compilation fails at line n2.
- B. Read Book
- C. Read E-Book
- D. Compilation fails at line n1.
- E. Compilation fails at line n3.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 83

Given the following class:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```



Which two changes would encapsulate this class and ensure that the area field is always equal to `length * height` whenever the Rectangle class is used?

- A. Call the setArea method at the end of the setHeight method.
- B. Call the setArea method at the beginning of the setHeight method.
- C. Call the setArea method at the end of the setLength method.
- D. Call the setArea method at the beginning of the setLength method.
- E. Change the setArea method to private.
- F. Change the area field to public.

**Correct Answer:** AE

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 84

Given the code fragment:

```
13. List colors = new ArrayList();  
14. colors.add("green");  
15. colors.add("red");  
16. colors.add("blue");  
17. colors.add("yellow");  
18. colors.remove(2);  
19. colors.add(3, "cyan");  
20. System.out.print(colors);
```

What is the result?

- A. (green, red, yellow, cyan)
- B. (green, blue, yellow, cyan)
- C. (green, red, cyan, yellow)
- D. An `IndexOutOfBoundsException` is thrown at runtime.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**





<https://vceplus.com/>

