# Oracle.Premium.1z0-808.by.VCEplus.142q

Number: 1z0-808
Passing Score: 800
Time Limit: 120 min
File Version: 8.2


**Exam Code: 1z0-808**
**Exam Name: Java SE 8 Programmer**
**Certification Provider: Oracle**
**Corresponding Certification: Oracle Certified Associate, Java SE 8 Programmer**
**Website:** www.vceplus.com
**Free Exam: https://vceplus.com/exam-1z0-808/**



Questions & Answers Exam Engine is rigorously checked before being put up for sale. We make sure there is nothing irrelevant in 1zo-808 exam products and you get latest questions. We strive to deliver the best 1z0-808 exam product for top grades in your first attempt

**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
**Google+ :** https://plus.google.com/+Vcepluscom
**LinkedIn :** https://www.linkedin.com/company/vceplus

**Exam A**

**QUESTION 1**
Given:

```
class Product {
    double price;
}

public class Test {
    public void updatePrice(Product product, double price) {
        price = price * 2;
        product.price = product.price + price;
    }
    public static void main(String[] args) {
        Product prt = new Product();
        prt.price = 200;
        double newPrice = 100;

        Test t = new Test();
        t.updatePrice(prt, newPrice);
        System.out.println(prt.price + " : " + newPrice);
    }
}
```



What is the result?

A. 200.0 : 100.0
B. 400.0 : 200.0
C. 400.0 : 100.0
D. Compilation fails.

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

**Explanation/Reference:**

Section: (none) Explanation
Explanation/Reference:

**QUESTION 2**
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. 10 Hello World!
B. Hello Universe!
C. Hello World
D. Compilation fails.

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



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

**QUESTION 3**
Given the code fragment:

```
public static void main(String[] args) {
    String date = LocalDate
                  .parse("2014-05-04")
                  .format(DateTimeFormatter.ISO_DATE_TIME);
    System.out.println(date);
}
```

What is the result?

A. May 04, 2014T00:00:00.000

B. 2014-05-04T00:00: 00. 000
C. 5/4/14T00:00:00.000
D. An exception is thrown at runtime.

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

**Explanation/Reference:**
Section: (none) Explanation
Explanation/Reference:
Reference: https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html (see predefined formatters)

**QUESTION 4**
Given the code fragment:

```
public static void main(String[] args) {
    Short s1 = 200;
    Integer s2 = 400;
    Long s3 = (long) s1 + s2;          //line n1
    String s4 = (String) (s3 * s2);    //line n2
    System.out.println("Sum is " + s4);
}
```



What is the result?

A. Sum is 600
B. Compilation fails at line n1.
C. Compilation fails at line n2.
D. A ClassCastException is thrown at line n1.
E. A ClassCastException is thrown at line n2.

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

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

**QUESTION 5**
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

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

**Explanation/Reference:**
Section: (none) Explanation
Explanation/Reference:
Explanation:
Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world. Reference: http://www.tutorialspoint.com/java/java_access_modifiers.htm



**QUESTION 6**
Given the code fragment:

```
abstract class Planet {
    protected void revolve() {            //line n1
    }

    abstract void rotate();               //line n2
}

class Earth extends Planet {
    void revolve() {                      //line n3
    }

    protected void rotate() {             //line n4
    }
}
```

Which two modifications, made independently, enable the code to compile?

A. Make the method at line n1 public.
B. Make the method at line n2 public.
C. Make the method at line n3 public.
D. Make the method at line n3 protected.
E. Make the method at line n4 public.

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

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

**QUESTION 7**
Given:

```
class Vehicle {
    String type = "4W";
    int maxSpeed = 100;

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

class Car extends Vehicle {
    String trans;

    Car(String trans) {          //line n1
        this.trans = trans;
    }

    Car(String type, int maxSpeed, String trans) {
        super(type, maxSpeed);
        this(trans);             //line n2
    }
}
```



And given the code fragment:

```
 7. Car c1 = new Car("Auto");
 8. Car c2 = new Car("4W", 150, "Manual");
 9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
```

What is the result?

A. 4W 100 Auto
   4W 150 Manual

B. Null 0 Auto
   4W 150 Manual

C. Compilation fails only at line n1

D. Compilation fails only at line n2
E. Compilation fails at both line n1 and line n2

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

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

**QUESTION 8**
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?

☐ 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();`

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



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

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

**QUESTION 9**
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 useElectricity 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:** AC
**Section: (none)**
**Explanation**

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

**QUESTION 10**
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:**
Section: (none) Explanation
Explanation/Reference:

**QUESTION 11**
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
}
```

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

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

**QUESTION 12**
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:** E
**Section: (none)**
**Explanation**

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

**QUESTION 13**
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:**
Section: (none) Explanation
Explanation/Reference:

**QUESTION 14**
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?

A. int e = 0; e < = 4; e++
B. int e = 0; e < 5; e + = 2
C. int e = 1; e < = 5; e + = 1
D. int e = 1; e < 5; e+ =2



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

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

**QUESTION 15**
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:**
Section: (none) Explanation
Explanation/Reference:

**QUESTION 16**
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.*;
```

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



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

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

**QUESTION 17**
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 o balance?

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

**QUESTION 18**
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:**
Section: (none) Explanation Explanation/Reference:

**QUESTION 19**
Given the code fragment:

```
public class Test{

    void readCard(int cardNo) throws Exception {
        System.out.println("Reading Card");
    }

    void checkCard(int cardNo) throws RuntimeException { // line n1
        System.out.println("Checking Card");
    }

    public static void main(String[] args) {
        Test ex = new Test();
        int cardNo = 12344;
        ex.checkCard(cardNo);                    //line n2
        ex.readCard(cardNo);                     //line n3
    }
}
```



What is the result?

A. Reading Card
   Checking Card
B. Compilation fails only at line n1.
C. Compilation fails only at line n2.
D. Compilation fails only at line n3.
E. Compilation fails at both line n2 and line n3.

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

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

**QUESTION 20**
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 1, 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;

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

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

**QUESTION 21**
Given the code fragment:

```
4. public static void main(String[] args) {
5.     boolean opt = true;
6.     switch (opt) {
7.         case true:
8.             System.out.print("True");
9.             break;
10.        default:
11.            System.out.print("***");
12.    }
13.    System.out.println("Done");
14. }
```

Which modification enables the code fragment to print TrueDone?

A. Replace line 5 With String result = "true"; Replace line 7 with case "true":
B. Replace line 5 with boolean opt = l; Replace line 7 with case 1=
C. At line 9, remove the break statement.
D. Remove the default section.



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

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

**QUESTION 22**
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:**
Section: (none) Explanation
Explanation/Reference:

**QUESTION 23**
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:**
Section: (none) Explanation
Explanation/Reference:

**QUESTION 24**
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.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. 91 98
99 100 null null null
B. 91 98
99100 101 102 103
C. Compilation rails.
D. A NullPointerException is thrown at runtime.

E. An ArrayIndexOutOfBoundsException is thrown at runtime.

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

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

**QUESTION 25**
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?

☐ 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 D
E. Option E

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

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

Explanation/Reference:

**QUESTION 26**
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:**
Section: (none) Explanation
Explanation/Reference:

**QUESTION 27**
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:**
Section: (none) Explanation
Explanation/Reference:

**QUESTION 28**
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:**
Section: (none) Explanation
Explanation/Reference:

**QUESTION 29**
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?

A. int[] array n= new int[2];
B. int[] array; array = int[2];
C. int array = new int[2];
D. int array [2] ;

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



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

**QUESTION 30**
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



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

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

**QUESTION 31**
Which three are advantages of the Java exception mechanism?

A. Improves the program structure because the error handling code is separated from the normal program function
B. Provides a set of standard exceptions that covers all the possible errors
C. Improves the program structure because the programmer can choose where to handle exceptions
D. Improves the program structure because exceptions must be handled in the method in which they occurred
E. Allows the creation of new exceptions that are tailored to the particular program being created

**Correct Answer:** ACD
**Section: (none)**

**Explanation**

**Explanation/Reference:**
Section: (none) Explanation Explanation/Reference:
Reference: http://javajee.com/introduction-to-exceptions-in-java

**QUESTION 32**
Given the code from the Greeting.Java file:

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

Which set of commands prints Hello Duke in the console?

A)
```
javac Greeting
java Greeting Duke
```

B)
```
javac Greeting.java Duke
java Greeting
```

C)
```
javac Greeting.java
java Greeting Duke
```

D)
```
javac Greeting.java
java Greeting.class Duke
```



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

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