

Oracle

Exam 1z0-808

Java SE 8 Programmer I

Version: 6.0

[Total Questions: 236]

Question No : 1

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

Answer: A

Question No : 2

Given:

```
public class Case {  
    public static void main(String[] args) {  
        String product = "Pen";  
        product.toLowerCase ();  
        product.concat (" BOX".toLowerCase ());  
        System.out.print (product.substring (4, 6));  
    }  
}
```

What is the result?

- A. box
- B. nbo
- C. bo
- D. nb
- E. An exception is thrown at runtime

Answer: E

Question No : 3

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

Answer: A,C,E

Question No : 4

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 + " " + 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?

- A. Jesse 25
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

Answer: D

Question No : 5

Given:

```
class Mid {
    public int findMid(int n1, int n2) {
        return (n1 + n2) / 2;
    }
}
```

```
}  
  
public class Calc extends Mid {  
  
public static void main(String[] args) {  
  
int n1 = 22, n2 = 2;  
  
// insert code here  
  
System.out.print(n3);  
  
}  
  
}
```

Which two code fragments, when inserted at // insert code here, enable the code to compile and print 12?

- A. `Calc c = new Calc();`
`int n3 = c.findMid(n1,n2);`
- B. `int n3 = super.findMid(n1,n3);`
- C. `Calc c = new Mid();`
`int n3 = c.findMid(n1, n2);`
- D. `Mid m1 = new Calc();`
`int n3 = m1.findMid(n1, n2);`
- E. `int n3 = Calc.findMid(n1, n2);`

Answer: A,D

Explanation:

Incorrect:

Not B: circular definition of n3.

Not C: Compilation error. line `Calc c = new Mid();`

required: Calc

found: Mid

Not E: Compilation error. line `int n3 = Calc.findMid(n1, n2);`

non-static method `findMid(int,int)` cannot be referenced from a static context

Question No : 6

Given the code fragment:

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

- A) date1 = 2014-06-20
date2 = 2014-06-20
date3 = 2014-06-20
- B) date1 = 06/20/2014
date2 = 2014-06-20
date3 = Jun 20, 2014
- C) Compilation fails.
- D) A DateParseException is thrown at runtime.

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

Answer: A

Question No : 7

Given the code fragment:

```
int[] lst = {1, 2, 3, 4, 5, 4, 3, 2, 1};
int sum = 0;
for (int frnt = 0, rear = lst.length - 1;
     frnt < 5 && rear >= 5;
     frnt++, rear--) {
    sum = sum + lst[frnt] + lst[rear];
}
System.out.print(sum);
```

What is the result?

- A. 20
- B. 25
- C. 29
- D. Compilation fails
- E. AnArrayIndexOutOfBoundsException is thrown at runtime

Answer: A

Question No : 8

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.

Answer: D

Explanation:

java.time.temporal.UnsupportedTemporalTypeException: Unsupported field: HourOfDay

Question No : 9

Given the code fragment:

```
public static void main(String[] args) {  
    double discount = 0;  
    int qty = Integer.parseInt(args[0]);  
    //line n1;  
}
```

And given the requirements:

- ✍ If the value of the qty variable is greater than or equal to 90, discount = 0.5
- ✍ If the value of the qty variable is between 80 and 90, discount = 0.2

Which two code fragments can be independently placed at line n1 to meet the requirements?

- A)

```
if (qty >= 90) { discount = 0.5; }  
if (qty > 80 && qty < 90) { discount = 0.2; }
```
- B)

```
discount = (qty >= 90) ? 0.5 : 0;  
discount = (qty > 80) ? 0.2 : 0;
```
- C)

```
discount = (qty >= 90) ? 0.5 : (qty > 80) ? 0.2 : 0;
```
- D)

```
if (qty > 80 && qty < 90) {  
    discount = 0.2;  
} else {  
    discount = 0;  
}  
if (qty >= 90) {  
    discount = 0.5;  
} else {  
    discount = 0;  
}
```
- E)

```
discount = (qty > 80) ? 0.2 : (qty >= 90) ? 0.5 : 0;
```

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

Answer: A,C

Question No : 10

Given:

```
public class Circle {
    double radius;
    public double area;
    public Circle(double r) { radius = r; }
    public double getRadius() { return radius; }
    public void setRadius(double r) { radius = r; }
    public double getArea() { return /* ??? */; }
}

class App {
    public static void main(String[] args) {
        Circle c1 = new Circle(17.4);
        c1.area = Math.PI * c1.getRadius() * c1.getRadius();
    }
}
```

The class is poorly encapsulated. You need to change the circle class to compute and return the area instead.

Which two modifications are necessary to ensure that the class is being properly encapsulated?

- A. Remove the area field.
- B. Change the getArea() method as follows:
`public double getArea () { return Match.PI * radius * radius; }`
- C. Add the following method:
`public double getArea () {area = Match.PI * radius * radius; }`
- D. Change the caccess modifier of the SerRadius () method to be protected.

Answer: B,D

Question No : 11

Given the code fragment:

```
12. int row = 10;
13. for ( ; row > 0 ; ) {
14.     int col = row;
15.     while (col >= 0) {
16.         System.out.print(col + " ");
17.         col -= 2;
18.     }
19.     row = row / col;
20. }
```

What is the result?

- A. 10 8 6 4 2 0
- B. 10 8 6 4 2
- C. AnArithmeticException is thrown at runtime
- D. The program goes into an infinite loop outputting: 10 8 6 4 2 0. . .
- E. Compilation fails

Answer: B

Question No : 12

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?

- A. `int f = ps.indexOf {new patient ("Mike")};`
- B. `int f = ps.indexOf (patient("Mike"));`
- C. `patient p = new Patient ("Mike");`
`int f = pas.indexOf(P)`
- D. `int f = ps.indexOf(p2);`

Answer: C