DUMPSARENA

# Java SE 8 Programmer I

## Oracle 1z0-808

## Version Demo

**Total Demo Questions: 15**

**Total Premium Questions: 226**

**Buy Premium PDF**

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

## QUESTION NO: 1

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 D

**E.** Option E

**ANSWER: A C**

**QUESTION NO: 2**

Given:

```
public class Test {
    public static final int MIN = 1;
    public static void main(String[] args) {
        int x = args.length;
        if(checkLimit(x)){      // line n1
            System.out.println("Java SE");
        } else {
            System.out.println("Java EE");
        }
    }
    public static boolean checkLimit(int x) {
        return (x >= MIN) ? true : false;
    }
}
```

And given the commands:

```
javac Test.java
java Test 1
```

What is the result?

**A.** Java SE

**B.** Java EE

**C.** Compilation fails at line n1.

**D.** A NullPointerException is thrown at runtime.

**ANSWER: B**

**QUESTION NO: 3**

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;

**ANSWER: C E**

**QUESTION NO: 4**

Given:

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

Given the code fragment:

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

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

**A.** At line n1 insert: import clothing.Shirt;
At line n2 insert: String color = Shirt.getColor();

**B.** At line n1 insert: import clothing;
At line n2 insert: String color = Shirt.getColor();

**C.** At line n1 insert: import static clothing.Shirt.getColor; At line n2 insert: String color = getColor();

**D.** At line n1 no changes required.
At line n2 insert: String color = Shirt.getColor();

**E.** At line n1 insert: import Shirt;
At line n2 insert: String color = Shirt.getColor();

**ANSWER: D**

## QUESTION NO: 5

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 main(String[] args) {
        doSum(10, 20);
        doSum(10.0, 20.0);
    }
}
```

What is the result?

A.
```
float sum is 30.0
double sum is 30.0
```

B.
```
double sum is 30.0
float sum is 30.0
```

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

**ANSWER: A**

**QUESTION NO: 6**

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. System.out.println(tiger.type + " " + tiger.maxSpeed + " " + tiger.bounds);
```

and this output: Canine 60 Long

Feline 80 Short

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

**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;
```

**ANSWER: A E**

**QUESTION NO: 7**

Given this class:

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

And given this 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.** acct.setAmount(-acct.getAmount());

**B.** acct.amount = 0;

**C.** acct.setAmount(0);

**D.** acct.getAmount() = 0;

**E.** this.amount = 0;

**F.** acct.changeAmount(0);

**G.** acct.changeAmount(-acct.amount);

**ANSWER: B D F**

## QUESTION NO: 8

Given:

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

Which two code fragments can be inserted at line n1? (Choose two.)

**A.** String str = “Java”;

**B.** for(int iVal = 0; iVal <=5; iVal++){}

**C.** Test() {}

**D.** package p1;

**E.** import java.io.*;

**ANSWER: A D**

## QUESTION NO: 9

Which three are advantages of the Java exception mechanism? (Choose three.)

**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 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 customized to the particular program being created

**ANSWER: A C E**

**Explanation:**

Reference: http://javajee.com/introduction-to-exceptions-in-java

**QUESTION NO: 10**

Given this code for the classes MyException and Test:

```
public class MyException extends RuntimeException {}

public class Test {
    public static void main(String[] args) {
       try {
           method1();
       }
       catch (MyException ne) {
           System.out.print("A");
       }
    }
    public static void method1() { // line n1
        try {
           throw 3 > 10 ? new MyException() : new IOException();
       }
        catch(IOException ie) {
            System.out.println("I");
        }
       catch (Exception re) {
           System.out.print("B");
       }
    }
}
```

What is the result?

**A.** A

**B.** AB

**C.** A compile time error occurs at line n1.

**D.** B

**E.** I

**ANSWER: E**

**QUESTION NO: 11**

Given:

```
public class Test {

    public static void main(String[] args) {

        String[][] chs = new String[5][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.** 97 98
99 100 null null null

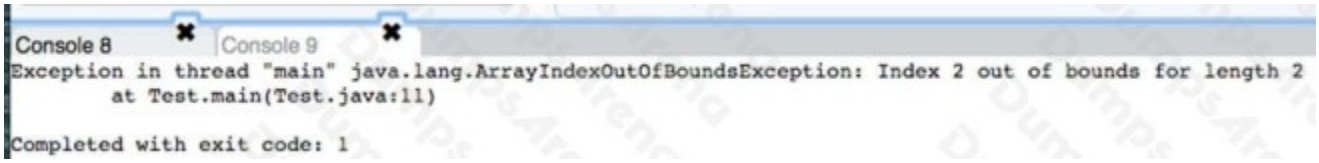
**B.** 97 98
99 100 101 102 103C. Compilation fails.

**C.** A NullPointerException is thrown at runtime.

**D.** An ArrayIndexOutOfBoundsException is thrown at runtime.

**ANSWER: D**

**Explanation:**




## QUESTION NO: 12

Given the code fragment:

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

Which modification enables the code to print 54321?

**A.** Replace line 6 with System.out.print (--x);

**B.** At line 7, insert x --;

**C.** Replace line 5 with while (is Available(--x)) {

**D.** Replace line 12 with return (x > 0) ? false : true;

**ANSWER: C**

## QUESTION NO: 13

Given the code fragment:

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

What is the result?

**A.** 1 2 3 0 0
1 2 3 0 0

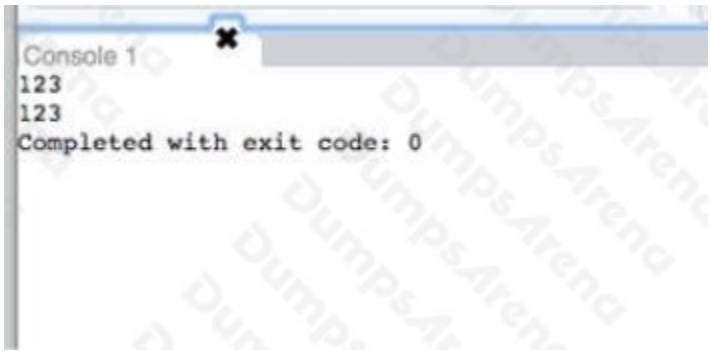
**B.** An Exception is thrown at run time.

**C.** 1 2 3 0 0
1 2 3

**D.** 1 2 3
1 2 3

## ANSWER: D

**Explanation:**




## QUESTION NO: 14

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.

**ANSWER: B E**

**QUESTION NO: 15**

Given the code fragment:

```
String str = "Sweet Sweat";
String str2 = str.trim().charAt(6) + "" +str.indexOf("Sw",1);
System.out.println(str2);
```

What is the result?

**A.** S 6

**B.** S 5

**C.** s-1

**D.** w 7

**ANSWER: C**

**Explanation:**

```
16 public class Shop {
17     public static void main(String[] args) {
18     String str = "Sweet sweat";
19     String str2 = str.trim().charAt(6) + "" +str.indexOf("Sw",1);
20     System.out.println(str2);
21     }
22 }
```

Result

**CPU Time: 0.27 sec(s), Memory: 35780 kilobyte(s)**

```
s-1
```