Vendor: Oracle

Exam Code: 1Z0-803

Exam Name: Java SE 7 Programmer I

Version: Demo

QUESTION NO: 1

Given the code fragment:

```
int [] [] array2D = {{0, 1, 2}, {3, 4, 5, 6}};

system.out.print (array2D[0].length+ "" );
system.out.print(array2D[1].getClass(). isArray() + "");
system.out.println (array2D[0][1]);
```

What is the result?

A. 3false1
B. 2true3
C. 2false3
D. 3true1
E. 3false3
F. 2true1
G. 2false1

Answer: D
Explanation: The length of the element with index 0, {0, 1, 2}, is 3. Output: 3
The element with index 1, {3, 4, 5, 6}, is of type array. Output: true
The element with index 0, {0, 1, 2} has the element with index 1: 1. Output: 1

QUESTION NO: 2

View the exhibit:

```
public class Student {
 public String name = "";
 public int age = 0;
 public String major = "Undeclared";
 public boolean fulltime = true;
```

```
 public void display() {
 System.out.println("Name: " + name + " Major: " + major);
 }
public boolean isFullTime() {
 return fulltime;
}
}
```

Given:

```
Public class TestStudent {
Public static void main(String[] args) {
Student bob = new Student ();
Student jian = new Student();

bob.name = "Bob";
bob.age = 19;
jian = bob; jian.name = "Jian";
System.out.println("Bob's Name: " + bob.name);
}
}
```

What is the result when this program is executed?

A. Bob's Name: Bob
B. Bob's Name: Jian
C. Nothing prints
D. Bob’s name

Answer: B
Explanation: After the statement jian = bob; the jian will reference the same object as bob.

QUESTION NO: 3

Given the code fragment:

```
String valid = "true";
if (valid) System.out.println (“valid”);
else system.out.println ("not valid");
```

What is the result?

A. Valid
B. not valid
C. Compilation fails
D. An IllegalArgumentException is thrown at run time

Answer: C
Explanation: In segment 'if (valid)' valid must be of type boolean, but it is a string.
This makes the compilation fail.

QUESTION NO: 4

Given:

```
public class ScopeTest {
int z;
public static void main(String[] args){
 ScopeTest myScope = new ScopeTest();
 int z = 6;
 System.out.println(z);
 myScope.doStuff();
 System.out.println(z);
 System.out.println(myScope.z);
 }
```

```
void doStuff() {
 int z = 5;
 doStuff2();
 System.out.println(z);
}
void doStuff2() {
 z=4;
}
}
```

What is the result?

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

Answer: A
Explanation: Within main z is assigned 6. z is printed. Output: 6
Within doStuff z is assigned 5.DoStuff2 locally sets z to 4 (but MyScope.z is set to 4), but in Dostuff z is still 5. z is printed. Output: 5
Again z is printed within main (with local z set to 6). Output: 6
Finally MyScope.z is printed. MyScope.z has been set to 4 within doStuff2(). Output: 4

QUESTION NO: 5

Which two are valid instantiations and initializations of a multi dimensional array?

A. int [] [] array 2D = { { 0, 1, 2, 4} {5, 6}};
B. int [] [] array2D = new int [2] [2];
array2D[0] [0] = 1;
array2D[0] [1] = 2;
array2D[1] [0] = 3;
array2D[1] [1] = 4;
C. int [] [] [] array3D = {{0, 1}, {2, 3}, {4, 5}};

D. int [] [] [] array3D = new int [2] [2] [2];
array3D [0] [0] = array;
array3D [0] [1] = array;
array3D [1] [0] = array;
array3D [0] [1] = array;
E. int [] [] array2D = {0, 1};

Answer: B,D
Explanation: In the Java programming language, a multidimensional array is simply an array whose components are themselves arrays.

QUESTION NO: 6

An unchecked exception occurs in a method dosomething()

Should other code be added in the dosomething() method for it to compile and execute?

A. The Exception must be caught
B. The Exception must be declared to be thrown.
C. The Exception must be caught or declared to be thrown.
D. No other code needs to be added.

Answer: D
Explanation: Because the Java programming language does not require methods to catch or to specify unchecked exceptions (RuntimeException,Error, and their subclasses), programmers may be tempted to write code that throws only unchecked exceptions or to make all their exception subclasses inherit fromRuntimeException. Both of these shortcuts allow programmers to write code without bothering with compiler errors and without bothering to specify or to catch any exceptions. Although this may seem convenient to the programmer, it sidesteps the intent of thecatchorspecifyrequirement and can cause problems for others using your classes.

QUESTION NO: 7

Given the code fragment:

int b = 4;

```
b -- ;

System.out.println (-- b);

System.out.println(b);
```

What is the result?

A. 2 2
B. 1 2
C. 3 2
D. 3 3

Answer: A
Explanation: Variable b is set to 4.
Variable b is decreased to 3.
Variable b is decreased to 2 and then printed. Output: 2
Variable b is printed. Output: 2

QUESTION NO: 8

Given the code fragment:

```
interface SampleClosable {

public void close () throws java.io.IOException;

}
```

Which three implementations are valid?

A.
```
public class Test implements SampleCloseable {
public void close() throws java.io.IOException {
/ / do something
}
}
```
B.
```
public class Test implements SampleCloseable {
public void close() throws Exception {
/ / do something
}
}
```

C. public class Test implements SampleCloseable {
public void close() throws java.io.FileNotFoundException {
/ / do something
}
}
D. public class Test extends SampleCloseable {
public void close() throws java.IO.IOException {
/ / do something
}
}
E. public class Test implements SampleCloseable {
public void close()
/ / do something
}
}

Answer: A,C,E

Explanation: A: Throwing the same exception is fine.

C: Using a subclass of java.io.IOException (here java.io.FileNotFoundException) is fine

E: Not using a throw clause is fine.

QUESTION NO: 9

Given the code fragment:

Int [] [] array = {{0}, {0, 1}, {0, 2, 4}, {0, 3, 6, 9}, {0, 4, 8, 12, 16}};

Systemout.printIn(array [4] [1]);

System.out.printIn (array) [1][4]);

int [] [] array = {{0}, {0, 1}, {0, 2, 4}, {0, 3, 6, 9}, {0, 4, 8, 12, 16}};

System.out.println(array [4][1]);

System.out.println(array) [1][4]);

What is the result?

A. 4 Null
B. Null 4
C. An IllegalArgumentException is thrown at run time
D. 4 An ArrayIndexOutOfBoundException is thrown at run time

Answer: D

Explanation: The first println statement, System.out.println(array [4][1]);, works fine. It selects the element/array with index 4, {0, 4, 8, 12, 16}, and from this array it selects the element with index 1, 4. Output: 4

The second println statement, System.out.println(array) [1][4]);, fails. It selects the array/element with index 1, {0, 1}, and from this array it try to select the element with index 4. This causes an exception.

Output:
4
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 4

QUESTION NO: 10

Given:

```
public class DoCompare1 {

public static void main(String[] args) {

 String[] table = {"aa", "bb", "cc"};

 for (String ss: table) {

 int ii = 0;

 while (ii < table.length) {

 System.out.println(ss + ", " + ii);

 ii++;

 }

}
```

How many times is 2 printed as a part of the output?

A. Zero
B. Once
C. Twice
D. Thrice

E. Compilation fails.

Answer: D

Explanation: The for statement, for (String ss: table), is executed one time for each of the three elements in table. The while loop will print a 2 once for each element.

Output:
aa, 0
aa, 1
aa, 2
bb, 0
bb, 1
bb, 2
cc, 0
cc, 1
cc, 2

QUESTION NO: 11

Given:

```
import java.io.IOException;

public class Y {
 public static void main(String[] args) {
 try {
 doSomething();
 }
 catch (RuntimeException e) {
 System.out.println(e);
 }
 }
 static void doSomething() {
```

```
if (Math.random() > 0.5) throw new IOException();

throw new RuntimeException();

}

}
```

Which two actions, used independently, will permit this class to compile?

A. Adding throws IOException to the main() method signature
B. Adding throws IOException to the doSoomething() method signature
C. Adding throws IOException to the main() method signature and to the dosomething() method
D. Adding throws IOException to the dosomething() method signature and changing the catch argument to IOException
E. Adding throws IOException to the main() method signature and changing the catch argument to IOException

Answer: C,D
Explanation: The IOException must be caught or be declared to be thrown.
We must add a throws exception to the doSomething () method signature (static void doSomething() throws IOException).
Then we can either add the same throws IOException to the main method (public static void main(String[] args) throws IOException), or change the catch statement in main to IOException.

QUESTION NO: 12

Given:

```
class X {

String str = "default";

X(String s) { str = s;}

void print () { System.out.println(str); }

public static void main(String[] args) {

new X("hello").print();

}
```

```
}
```

What is the result?

A. hello
B. default
C. Compilation fails
D. The program prints nothing
E. An exception is thrown at run time

Answer: A
Explanation: The program compiles fine.
The program runs fine.
The output is: hello

QUESTION NO: 13

Given:

```
public class SampleClass {
 public static void main(String[] args) {
AnotherSampleClass asc = new AnotherSampleClass();
SampleClass sc = new SampleClass();
// TODO code application logic here
 }
}
class AnotherSampleClass extends SampleClass {
 }
```

Which statement, when inserted into line "// TODO code application logic here ", is valid change?

A. asc = sc;
B. sc = asc;
C. asc = (object) sc;
D. asc = sc.clone ()

Answer: B
Explanation: Works fine.

QUESTION NO: 14

Given the code fragment:

System.out.println("Result: " + 2 + 3 + 5);

System.out.println("Result: " + 2 + 3 * 5);

What is the result?

A. Result: 10
Result: 30
B. Result: 10
Result: 25
C. Result: 235
Result: 215
D. Result: 215
Result: 215
E. Compilation fails

Answer: C
Explanation: First line:
System.out.println("Result: " + 2 + 3 + 5);
String concatenation is produced.

Second line:
System.out.println("Result: " + 2 + 3 * 5);
3*5 is calculated to 15 and is appended to string 2. Result 215.

The output is:
Result: 235
Result: 215

Note #1:
To produce an arithmetic result, the following code would have to be used:
System.out.println("Result: " + (2 + 3 + 5));
System.out.println("Result: " + (2 + 1 * 5));
run:

Result: 10
Result: 7

Note #2:
If the code was as follows:

System.out.println("Result: " + 2 + 3 + 5");
System.out.println("Result: " + 2 + 1 * 5");

The compilation would fail. There is an unclosed string literal, 5", on each line.

QUESTION NO: 15

Which code fragment is illegal?

A. class Base1 {
abstract class Abs1 { }}
B. abstract class Abs1 {
void doit () { }
}
C. class Basel {
abstract class Abs1 extends Basel {
D. abstract int var1 = 89;

Answer: D
Explanation: The abstract keyword cannot be used to declare an int variable.

The abstract keyword is used to declare a class or method to beabstract[3]. An abstract method has no implementation; all classes containing abstract methods must themselves be abstract, although not all abstract classes have abstract methods.

QUESTION NO: 16

Given the code fragment:

int a = 0;

```
a++;

System.out.println(a++);

System.out.println(a);
```

What is the result?

A.
1
2
B.
0
1
C.
1
1
D.
2
2

Answer: A
Explanation: The first println prints variable a with value 1 and then increases the variable to 2.

QUESTION NO: 17

Given:

```
public class x{

public static void main (string [] args){

String theString = "Hello World";

System.out.println(theString.charAt(11));

 }

}
```

What is the result?

A. There is no output
B. d is output
C. A StringIndexOutOfBoundsException is thrown at runtime
D. An ArrayIndexOutOfBoundsException is thrown at runtime
E. A NullPointException is thrown at runtime
F. A StringArrayIndexOutOfBoundsException is thrown at runtime

Answer: C
Explanation: There are only 11 characters in the string "Hello World". The code theString.charAt(11) retrieves the 12th character, which does not exist. A StringIndexOutOfBoundsException is thrown.
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 11

QUESTION NO: 18

Given a java source file:

```
class x {
x () {}
private void one () {}
}
public class Y extends x {
Y () {}
private void two () {one();}
public static void main (string [] args) {
new Y().two ();
}
}
```

What changes will make this code compile?

A. adding the public modifier to the declaration of class x
B. adding the protected modifier to the x() constructor

C. changing the private modifier on the declaration of the one() method to protected
D. removing the Y () constructor
E. removing the private modifier from the two () method

Answer: C
Explanation: Using the private protected, instead of the private modifier, for the declaration of the one() method, would enable the two() method to access the one() method.

QUESTION NO: 19

Given:

#1

```
package handy.dandy;

public class KeyStroke {

 public void typeExclamation() {

 System.out.println("!")

 }

}
```

#2

```
package handy;  /*  Line 1 */

public class Greet {  /*  Line 2 */

 public static void main(String[] args) {  /*  Line 3 */

 String greeting = "Hello";  /*  Line 4 */

 System.out.print(greeting);  /*  Line 5 */

 Keystroke stroke = new Keystroke;  /*  Line 6 */

 stroke.typeExclamation();  /*  Line 7 */

 }  /*  Line 8 */

}  /*  Line 9 */
```

What three modifications, made independently, made to class greet, enable the code to compile and run?

A. Line 6 replaced with handy.dandy.keystroke stroke = new KeyStroke ( );
B. Line 6 replaced with handy.*.KeyStroke = new KeyStroke ( );
C. Line 6 replaced with handy.dandy.KeyStroke Stroke = new handy.dandy.KeyStroke();
D. import handy.*; added before line 1
E. import handy.dandy.*; added after line 1
F. import handy.dandy,KeyStroke; added after line 1
G. import handy.dandy.KeyStroke.typeException(); added before line 1

Answer: C,E,F

Explanation: Three separate solutions:

C: the full class path to the method must be stated (when we have not imported the package)

D: We can import the hold dandy class

F: we can import the specific method

QUESTION NO: 20

Given:

```
String message1 = "Wham bam!";

String message2 = new String("Wham bam!");

if (message1 == message2)

 System.out.println("They match");

if (message1.equals(message2))

 System.out.println("They really match");
```

What is the result?

A. They match
They really match
B. They really match
C. They match
D. Nothing Prints
E. They really match

They really match

Answer: B

Explanation: The strings are not the same objects so the == comparison fails. See note #1 below. As the value of the strings are the same equals is true. The equals method compares values for equality.

Note: #1 ==
Compares references, not values. The use of == with object references is generally limited to the following:
Comparing to see if a reference is null.
Comparing two enum values. This works because there is only one object for each enum constant.
You want to know if two references are to the same object.

QUESTION NO: 21

Given:

```
public class Speak {  /* Line 1 */
 public static void main(String[] args) {  /* Line 2 */
Speak speakIT = new Tell();  /* Line 3 */
Tell tellIt = new Tell();  /* Line 4 */
speakIT.tellItLikeItIs();  /* Line 5 */
(Truth)speakIt.tellItLikeItIs();  /* Line 6 */
((Truth)speakIt).tellItLikeItIs();  /* Line 7 */
tellIt.tellItLikeItIs();  /* Line 8 */
(Truth)tellIt.tellItLikeItIs();  /* Line 9 */
((Truth)tellIt).tellItLikeItIs();  /* Line 10 */
 }
}
class Tell extends Speak implements Truth {
```

```
 public void tellItLikeItIs() {
 System.out.println("Right on!");
 }
}
interface Truth { public void tellItLikeItIs()};
```

Which three lines will compile and output “right on!”?

A. Line 5
B. Line 6
C. Line 7
D. Line 8
E. Line 9
F. Line 10

Answer: C,D,F
Explanation:

QUESTION NO: 22

Given the code fragment:

```
String h1 = "Bob";
String h2 = new String ("Bob");
```

What is the best way to test that the values of h1 and h2 are the same?

A. if (h1 == h2)
B. if (h1.equals(h2))
C. if (h1 = = h2)
D. if (h1.same(h2))

Answer: B
Explanation: The equals method compares values for equality.

QUESTION NO: 23

Which two are valid declarations of a two-dimensional array?

A. int[][] array2D;
B. int[2][2] array2D;
C. int array2D[];
D. int[] array2D[];
E. int[][] array2D[];

Answer: A,D
Explanation: int[][] array2D; is the standard convention to declare a 2-dimensional integer array.

int[] array2D[]; works as well, but it is not recommended.

QUESTION NO: 24

Given the code fragment:

System.out.println ("Result:" +3+5);

System.out.println ("result:" + (3+5));

What is the result?

A. Result: 8
Result: 8
B. Result: 35
Result: 8
C. Result: 8
Result: 35
D. Result: 35
Result: 35

Answer: B
Explanation: In the first statement 3 and 5 are treated as strings and are simply concatenated.
In the first statement 3 and 5 are treated as integers and their sum is calculated.

QUESTION NO: 25

Given:

```
public class Main {
 public static void main(String[] args) throws Exception {
 doSomething();
 }
 private static void doSomething() throws Exception {
 System.out.println("Before if clause");
 if (Math.random() > 0.5) {
 throw new Exception();
 }
 System.out.println ("After if clause");
}
}
```

Which two are possible outputs?

A. Before if clause
Exception in thread “main” java.lang.Exception
At Main.doSomething (Main.java:8)
At Main.main (Main.java:3)
B. Before if clause
Exception in thread “main” java.lang.Exception
At Main.doSomething (Main.java:8)
At Main.main (Main.java:3)
After if clause
C. Exception in thread “main” java.lang.Exception
At Main.doSomething (Main.java:8)
At Main.main (Main.java:3)
D. Before if clause
After if clause

Answer: A,D

Explanation: The first println statement, System.out.println("Before if clause");, will always run. If Math.Random() > 0.5 then there is an exception. The exception message is displayed and the

program terminates.
If Math.Random() > 0.5 is false, then the second println statement runs as well.

QUESTION NO: 26

A method doSomething () that has no exception handling code is modified to trail a method that throws a checked exception. Which two modifications, made independently, will allow the program to compile?

A. Catch the exception in the method doSomething().
B. Declare the exception to be thrown in the doSomething() method signature.
C. Cast the exception to a RunTimeException in the doSomething() method.
D. Catch the exception in the method that calls doSomething().

Answer: A,B
Explanation: Valid Java programming language code must honor the Catch or Specify Requirement. This means that code that might throw certain exceptions must be enclosed by either of the following:

* A try statement that catches the exception. The try must provide a handler for the exception, as described in Catching and Handling Exceptions.
* A method that specifies that it can throw the exception. The method must provide a throws clause that lists the exception, as described in Specifying the Exceptions Thrown by a Method.

Code that fails to honor the Catch or Specify Requirement will not compile.

QUESTION NO: 27

Given the code fragment:

```
String color = "Red";

switch(color) {

case "Red":

System.out.println("Found Red");

case "Blue":
```

```
System.out.println("Found Blue");

break;

case "White":

System.out.println("Found White");

break;

default:

System.out.println("Found Default");

 }
```

What is the result?

A. Found Red
B. Found Red
Found Blue
C. Found Red
Found Blue
Found White
D. Found Red
Found Blue
Found White
Found Default

Answer: B
Explanation: As there is no break statement after the case "Red" statement the case Blue statement will run as well.

Note: The body of a switch statement is known as a switch block. A statement in the switch block can be labeled with one or more case or default labels. The switch statement evaluates its expression, then executes all statements that follow the matching case label.

Each break statement terminates the enclosing switch statement. Control flow continues with the first statement following the switch block. The break statements are necessary because without them, statements in switch blocks*fall through*: All statements after the matching case label are executed in sequence, regardless of the expression of subsequent case labels, until a break statement is encountered.

QUESTION NO: 28

Which two may precede the word "class" in a class declaration?

A. local
B. public
C. static
D. volatile
E. synchronized

Answer: B,C
Explanation: B: A class can be declared as public or private.

C: You can declare two kinds of classes: top-level classes and inner classes.
You define an inner class within a top-level class. Depending on how it is defined, an inner class can be one of the following four types:Anonymous, Local, Member and Nested top-level.
A nested top-level class is a member classes with astaticmodifier. A nested top-level class is just like any other top-level class except that it is declared within another class or interface. Nested top-level classes are typically used as a convenient way to group related classes without creating a new package.
The following is an example:

public class Main {
static class Killer {

QUESTION NO: 29

Which three are bad practices?

A. Checking for ArrayindexoutofBoundsException when iterating through an array to determine when all elements have been visited
B. Checking for Error and. If necessary, restarting the program to ensure that users are unaware problems
C. Checking for FileNotFoundException to inform a user that a filename entered is not valid
D. Checking for ArrayIndexoutofBoundsExcepcion and ensuring that the program can recover if one occur
E. Checking for an IOException and ensuring that the program can recover if one occurs

Answer: A,B,E
Explanation: A, E: Better to check if the index is within bounds.
B: Restarting the program would not be a good practice. It should be done as a last possibility

only.

QUESTION NO: 30

Given:

```
public class Bark {
 // Insert code here - Line 5
 public abstract void bark(); // Line 6
}  // Line 7
 // Line 8
 // Insert code here - Line 9
 public  void bark() {
System.out.println("woof");
 }
}
```

What code should be inserted?

A. 5.class Dog {
9. public class Poodle extends Dog {
B. 5. abstract Dog {
9. public class poodle extends Dog {
C. 5. abstract class Dog {
9. public class Poodle extends Dog {
D. 5. abstract Dog {
9. public class Poodle implements Dog {
E. 5. abstract Dog {
9. public class Poodle implements Dog {
F. 5. abstract class Dog {
9. public class Poodle implements Dog {

Answer: C

Explanation: Dog should be an abstract class. The correct syntax for this is: abstract class Dog { Poodle should extend Dog (not implement).

QUESTION NO: 31

Given:

class X {}

class Y {Y () {}}

class Z {z(int i) {} }

Which class has a default constructor?

A. X only
B. Y only
C. Z only
D. X and Y
E. Y and Z
F. X and Z
G. X, Y and Z

Answer: F
Explanation:

QUESTION NO: 32

Given:

Public static void main (String [] args) {

int a, b, c = 0;

int a, b, c;

int g, int h, int i, = 0;

int d, e, F;

int k, l, m; = 0;

Which two declarations will compile?

A. int a, b, c = 0;
B. int a, b, c;
C. int g, int h, int i = 0;
D. int d, e, F;
E. int k, l, m = 0;

Answer: A,D

QUESTION NO: 33

Given the code fragment:

```
int j=0, k =0;

for (int i=0; i < x; i++) {
do {
k=0;
while (k < z) {
k++;
System.out.print(k + " ");
}
System.out.println(" ");
j++;
} while (j< y);
System.out.println("----");
}
```

What values of x, y, z will produce the following result?

1 2 3 4

1 2 3 4

1 2 3 4

---

1 2 3 4

A. X = 4, Y = 3, Z = 2
B. X = 3, Y = 2, Z = 3
C. X = 2, Y = 3, Z = 3
D. X = 4, Y = 2, Z = 3
E. X = 2, Y = 3, Z = 4

Answer: E
Explanation: Z is for the innermost loop. Should print 1 2 3 4. So Z must be 4.
Y is for the middle loop. Should print three lines of 1 2 3 4. So Y must be set 3.
X is for the outmost loop. Should print 2 lines of ----. So X should be 2.

QUESTION NO: 34

Which statement initializes a stringBuilder to a capacity of 128?

A. StringBuilder sb = new String("128");
B. StringBuilder sb = StringBuilder.setCapacity(128);
C. StringBuilder sb = StringBuilder.getInstance(128);
D. StringBuilder sb = new StringBuilder(128);

Answer: D
Explanation: StringBuilder(int capacity)Constructs a string builder with no characters in it and an initial capacity specified by thecapacityargument.
Note: An instance of a StringBuilder is a mutable sequence of characters.
The principal operations on aStringBuilderare theappendandinsertmethods, which are overloaded so as to accept data of any type. Each effectively converts a given datum to a string and then appends or inserts the characters of that string to the string builder. Theappendmethod always adds these characters at the end of the builder; theinsertmethod adds the characters at a specified point.

QUESTION NO: 35

Given:

```
public class DoCompare4 {

public static void main(String[] args) {

String[]  table = {"aa", "bb", "cc"};

int ii =0;

do

while (ii < table.length)

System.out.println(ii++);

while (ii < table.length);

}

}
```

What is the result?

A. 0
B. 0
1
2
C. 0
1
2
0
1
2
0
1
2
D. Compilation fails

Answer: B
Explanation: table.length is 3. So the do-while loop will run 3 times with ii=0, ii=1 and ii=2.
The second while statement will break the do-loop when ii = 3.
Note:The Java programming language provides ado-whilestatement, which can be expressed as follows:

```
do {
statement(s)
} while (expression);
```

QUESTION NO: 36

A method is declared to take three arguments. A program calls this method and passes only two arguments. What is the result?

A. Compilation fails.
B. The third argument is given the value null.
C. The third argument is given the value void.
D. The third argument is given the value zero.
E. The third argument is given the appropriate false value for its declared type.
F. An exception occurs when the method attempts to access the third argument.

Answer: A
Explanation: The problem is noticed at build/compile time. At build you would receive an error message like:

required: int,int,int
found: int,int

QUESTION NO: 37

Given the fragment:

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

System.arraycopy (array, 2, array, 1, 2);

System.out.print (array [1]);

System.out.print (array[4]);
```

What is the result?

A. 14
B. 15
C. 24

D. 25
E. 34
F. 35

Answer: F
Explanation: The two elements 3 and 4 (starting from position with index 2) are copied into position index 1 and 2 in the same array.
After the arraycopy command the array looks like:
{1, 3, 4, 4, 5};

Then element with index 1 is printed: 3
Then element with index 4 is printed: 5
Note:The System class has an arraycopy method that you can use to efficiently copy data from one array into another:
public static void arraycopy(Object src, int srcPos,
Object dest, int destPos, int length)
The two Object arguments specify the array to copy*from*and the array to copy*to*. The three int arguments specify the starting position in the source array, the starting position in the destination array, and the number of array elements to copy.

QUESTION NO: 38

Given the following code fragment:

```
if (value >= 0) {

if (value != 0)

System.out.print("the ");

else

System.out.print("quick ");

if (value < 10)

 System.out.print("brown ");

if (value > 30)

System.out.print("fox ");

else if (value < 50)
```

```
System.out.print("jumps ");

else if (value < 10)

System.out.print("over ");

else

System.out.print("the ");

if (value > 10)

System.out.print("lazy ");

} else {

System.out.print("dog ");

}

System.out.print("… ");

}
```

What is the result if the integer value is 33?

A. The fox jump lazy …
B. The fox lazy …
C. Quick fox over lazy …
D. Quick fox the ….

Answer: B
Explanation: 33 is greater than 0.
33 is not equal to 0.
the is printed.
33 is greater than 30
fox is printed
33 is greater then 10 (the two else if are skipped)
lazy is printed
finally … is printed.

QUESTION NO: 39

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

Answer: A,C,E

Explanation: A: The error handling is separated from the normal program logic.

C: You have some choice where to handle the exceptions.

E: You can create your own exceptions.

QUESTION NO: 40

Given:

```
public class MyFor3 {

public static void main(String [] args) {

int [] xx = null;

System.out.println(xx);

}

}
```

What is the result?

A. null
B. compilation fails
C. Java.lang.NullPointerException
D. 0

Answer: A

Explanation: An array variable (here xx) can very well have the null value.

Note:
Null is the reserved constant used in Java to represent a void reference i.e a pointer to nothing. Internally it is just a binary 0, but in the high level Java language, it is a magic constant, quite distinct from zero, that internally could have any representation.

Trying our product !

★ **100%** Guaranteed Success

★ **100%** Money Back Guarantee

★ **365 Days** Free Update

★ **Instant Download** After Purchase

★ **24x7** Customer Support

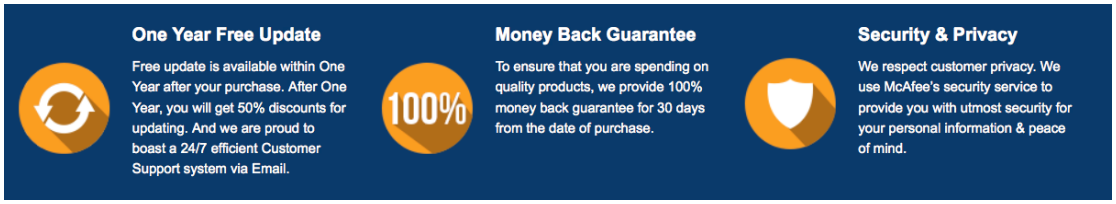
★ Average **99.9%** Success Rate

★ More than **69,000** Satisfied Customers Worldwide

★ Multi-Platform capabilities - **Windows, Mac, Android, iPhone, iPod, iPad, Kindle**

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket:



Guarantee & Policy | Privacy & Policy | Terms & Conditions