Oracle.certkiller.1Z0-803.v2014-09-20.by.BERNARDINA.158q

Exam Code: 1Z0-803
w
Exam Name: Java SE 7 Programmer I

CERTKILLER
slay your exams

Exam A

QUESTION 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

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
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 2
View the exhibit:

public class Student {

    public String name = "";

    public int age = 0;

```java
    public String major = "Undeclared";

    public boolean fulltime = true;


    public void display() {

        System.out.println("Name: " + name + " Major: " + major);
    }
    public boolean isFullTime() {

        return fulltime;

    }
}
```

Given:

```java
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

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation: After the statement jian = bob; the jian will reference the same object as bob.

QUESTION 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

Correct Answer: C
Section: (none)
Explanation

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

QUESTION 4
Given:

```java
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

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
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 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};

Correct Answer: BD
Section: (none)
Explanation

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

QUESTION 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.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
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 from RuntimeException. 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 the catch or specify requirement and can cause problems for others using your classes.

QUESTION 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

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
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 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
}
}
```

Correct Answer: ACE
Section: (none)
Explanation

Explanation/Reference:
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 9
Given the code fragment:

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

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

System.out.println (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

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

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

QUESTION 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.

Correct Answer: D
Section: (none)

Explanation

Explanation/Reference:
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 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

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:
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 12
Given:

```
lass 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

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Explanation: The program compiles fine.
The program runs fine.
The output is: hello

QUESTION 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 ()

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation: Works fine.

QUESTION 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

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
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 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;

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation: The abstract keyword cannot be used to declare an int variable.

The abstract keyword is used to declare a class or method to be abstract[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 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

Correct Answer: A
Section: (none)
Explanation

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

QUESTION 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

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
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:

QUESTION 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

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: CEF

Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: CDF
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 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))

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation: The equals method compares values for equality.

QUESTION 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[];

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
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 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().

Correct Answer: AB
Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
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 28
Which two may precede the word "class" in a class declaration?

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

Correct Answer: BC

Explanation/Reference:
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 a static modifier. 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 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

Correct Answer: ABD
Section: (none)
Explanation

Explanation/Reference:
Explanation: A, D: 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 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 {

Correct Answer: C
Section: (none)
Explanation

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

QUESTION 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

Correct Answer: F
Section: (none)
Explanation

Explanation/Reference:
Explanation: X an Z classes do not define constructors explicitly.

QUESTION 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 three 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;

Correct Answer: ABD
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 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

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
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 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);

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation: StringBuilder(int capacity) Constructs a string builder with no characters in it and an initial capacity specified by the capacity argument. Note: An instance of a StringBuilder is a mutable sequence of characters. The principal operations on a StringBuilder are the append and insert methods, 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. The append method always adds these characters at the end of the builder; the insert method adds the characters at a specified point.

QUESTION 35

Given:

```java
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
    1
B. 0
    1
    2
C. 0
    1
    2
    3
D. Compilation fails

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
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 a do-while statement, which can be expressed as follows:
do {
statement(s)
} while (expression);

QUESTION 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.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: F
Section: (none)
Explanation

Explanation/Reference:
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 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 ....

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: ACE
Section: (none)
Explanation

Explanation/Reference:
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 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

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
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.

QUESTION 41
Given:

```
public class Main
{
    public static void main(String[] args)
    {
        doSomething();
    }

    private static void doSomething()
    {
        doSomeThingElse();
    }

    private static void doSomeThingElse()
    {
        throw new Exception();
    }
}
```

Which approach ensures that the class can be compiled and run?

A. Put the throw new Exception() statement in the try block of try catch
B. Put the doSomethingElse() method in the try block of a try catch
C. Put the doSomething() method in the try block of a try catch
D. Put the doSomething() method and the doSomethingElse() method in the try block of a try catch

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Explanation: We need to catch the exception in the doSomethingElse() method.
Such as:
private static void doSomeThingElse() {
try {
throw new Exception();}
catch (Exception e)
{}
}

Note: One alternative, but not an option here, is the declare the exception in doSomeThingElse and catch it in the doSomeThing method.

QUESTION 42
Given:

```
public class ScopeTest1
{

    public static void main(String[] args)
    {

        doStuff(); // line x1

        int x1 = x2; // line x2

        int x2 = j; // line x3

    }

    static void doStuff() {

    System.out.println(j); // line x4

}

    static int j;

}
```

Which line causes a compilation error?

A. line x1

B. line x2

C. line x3

D. line x4

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation: The variable x2 is used before it has been declared.

QUESTION 43
Given:

```
class Overloading {

    void x(int i) {

        System.out.println("one");

    }

    void x(String s) {

        System.out.println("two");

    }

    void x(double d) {

        System.out.println("three");

    }

    public static void main(String[] args) {

        new Overloading().x(4.0);

    }

}
```

What is the result?

A. One
B. Two
C. Three
D. Compilation fails

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation: In this scenario the overloading method is called with a double/float value, 4.0. This makes the third overload method to run.

Note:
The Java programming language supports overloading methods, and Java can distinguish between methods with different method signatures. This means that methods within a class can have the same name if they have different parameter lists. Overloaded methods are differentiated by the number and the type of the arguments passed into the method.

QUESTION 44
Which declaration initializes a boolean variable?

A. boolean h = 1;
B. boolean k = 0;
C. boolean m = null;
D. boolean j = (1 < 5) ;

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation: The primitive type boolean has only two possible values: true and false. Here j is set to (1 <5), which evaluates to true.

QUESTION 45
Given:

public class Basic {

```
private static int letter;

public static int getLetter();

public static void Main(String[] args) {

    System.out.println(getLetter());

}

}
```

Why will the code not compile?

A. A static field cannot be private.
B. The getLetter method has no body.
C. There is no setletter method.
D. The letter field is uninitialized.
E. It contains a method named Main instead of main

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation: The getLetter() method needs a body public static int getLetter() { }; .

QUESTION 46
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();
    }
}
```

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

What three modifications are necessary to ensure that the class is being properly encapsulated?

A. Change the access modifier of the setradius () method to private
B. Change the getArea () method
    public double getArea () { return area; }
C. When the radius is set in the Circle constructor and the setRadius () method, recomputed the area and store it into the area field
D. Change the getRadius () method:
    public double getRadius () {
    area = Math.PI * radius * radius;
    return radius;}

Correct Answer: ABC
Section: (none)
Explanation

Explanation/Reference:
Explanation: A: There is no need to have SetRadius as public as the radius can be set through the Circle method.
B: We need to return the area in the GetArea method.
C: When the radius changes the Area must change as well.

Incorrect answer:
D: the GetRadius() method does not change the radius, so there is no need to recomputed the area.

QUESTION 47
Given a code fragment:

StringBuilder sb = new StringBuilder();

String h1 = "HelloWorld";

sb.append("Hello").append("world");

if (h1 == sb.toString()) {

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

}

if (h1.equals(sb.toString())) {

   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 is printed to the screen

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation: Strings can not be compared with the usual <, <=, >, or >= operators, and the == and != operators don't compare the characters in the strings. So the first if statement fails.

Equals works fine on strings. But it does not work here.The second if-statement also fails. The StringBuffer class does not override the equals method so it uses the equals method of Object. If a and b are two objects from a class which doesn't override equals, thena.equals(b) is the same as a == b

QUESTION 48

Given the following code:

public class Simple { /* Line 1 */

    public float price; /* Line 2 */

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

        Simple price = new Simple(); /* Line 4 */

        price = 4; /* Line 5 */

    } /* Line 6 */

} /* Line 7 */

What will make this code compile and run?

A. Change line 2 to the following:
   Public int price
B. Change line 4 to the following:
   int price = new simple ();
C. Change line 4 to the following:
   Float price = new simple ();
D. Change line 5 to the following:
   Price = 4f;
E. Change line 5 to the following:
   price.price = 4;
F. Change line 5 to the following:
   Price = (float) 4:
G. Change line 5 to the following:
   Price = (Simple) 4;
H. The code compiles and runs properly; no changes are necessary

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
Explanation: price.price =4; is correct, not price=4; The attribute price of the instance must be set, not the instance itself.

QUESTION 49
Given:

```
public class DoWhile {

    public static void main(String[] args) {

        int ii = 2;

        do {

            System.out.println(ii);

        } while (--ii);

    }

}
```

What is the result?

A. 1
B. 2
C. null
D. an infinite loop
E. compilation fails

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
Explanation: The line while (--ii); will cause the compilation to fail.
--ii is not a boolean value.
A correct line would be while (--ii>0);

QUESTION 50
You are writing a method that is declared not to return a value. Which two are permitted in the method body?

A. omission of the return statement

B. return null;

C. return void;

D. return;

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
Explanation: Any method declared void doesn't return a value. It does not need to contain a return statement, but it may do so. In such a case, a return statement can be used to branch out of a control flow block and exit the method and is simply used like this:
return;

QUESTION 51
Identify two benefits of using ArrayList over array in software development.

A. reduces memory footprint

B. implements the Collection API

C. is multi.thread safe

D. dynamically resizes based on the number of elements in the list

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
Explanation: ArrayList supports dynamic arrays that can grow as needed. In Java, standard arrays are of a fixed length. After arrays are created, they cannot grow or shrink, which means that you must know in advance how many elements an array will hold. But, sometimes, you may not know until run time precisely how large of an array you need. To handle this situation, the collections framework defines ArrayList. In essence, an ArrayList is a variable-length array of object references. That is, an ArrayList can dynamically increase or decrease in size. Array lists
are created with an initial size. When this size is exceeded, the collection is automatically enlarged. When objects are removed, the array may be shrunk.

QUESTION 52
Which three are valid types for switch?

A. int

B. float

C. double

D. Integer
E. String
F. Float

Correct Answer: ADE
Section: (none)
Explanation

Explanation/Reference:
Explanation: A switch works with the byte, short, char, and int primitive data types. It also works with enumerated types the String class, and a few special classes that wrap certain primitive types: Character, Byte, Short, and Integer.

QUESTION 53
Give:

public class MyFive {

   static void main(String[] args) {

      short ii;

      short jj = 0;

      for (ii = kk; ii > 6; ii -= 1) { // line x //

         jj++;

      }

      System.out.println("jj = " + jj);

   }
}

What value should replace KK in line x to cause jj = 5 to be output?

A. -1
B. 1
C. 5
D. 8

E. 11

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
Explanation: We need to get jj to 5. It is initially set to 0. So we need to go through the for loop 5 times. The for loops ends when ii > 6 and ii decreases for every loop. So we need to initially set ii to 11. We set kk to 11.

QUESTION 54
Given the code fragment:

Boolean b1 = true;

Boolean b2 = false;

int 1 = 0;

while (foo) {}

Which one is valid as a replacement for foo?

A. b1.compareTo(b2)
B. i = 1
C. i == 2? -1:0
D. "foo".equals("bar")

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation: equals works fine on strings. equals produces a Boolean value.

QUESTION 55
Given:

public class SuperTest
{
    public static void main(String[] args)

```java
    {
        statement1
        statement2
        statement3
    }
}

class Shape
{
    public Shape()
    {
        System.out.println("Shape: constructor");
    }

    public void foo()
    {
        System.out.println("Shape: foo");
    }
}

class Square extends Shape
{
    public Square()
    {
        super();
    }

    public Square(String label)
    {
        System.out.println("Square: constructor");
    }

    public void foo()
    {
        super.foo();
    }

    public void foo(String label)
    {
        System.out.println("Square: foo");
    }
}
```

What should statement1, statement2, and statement3, be respectively, in order to produce the result:

Shape: constructor
Square: foo
Shape: foo

A. Square square = new Square ("bar");
    square.foo ("bar");
    square.foo();
B. Square square = new Square ("bar");
    square.foo ("bar");
    square.foo ("bar");
C. Square square = new Square ();
    square.foo ();
    square.foo(bar);
D. Square square = new Square ();
    square.foo ();
    square.foo("bar");
E. Square square = new Square ();
    square.foo ();
    square.foo ();
F. Square square = new Square ();
    square.foo("bar");
    square.foo ();

Correct Answer: F
Section: (none)
Explanation

Explanation/Reference:

QUESTION 56
Give:

Public Class Test {

}

Which two packages are automatically imported into the java source file by the java compiler?

A. Java.lang
B. Java.awt
C. Javax.net
D. Java.*
E. The package with no name

Correct Answer: AE
Section: (none)
Explanation

Explanation/Reference:
Explanation: For convenience, the Java compiler automatically imports three entire packages for each source file: (1) the package with no name, (2) the java.lang package, and (3) the current package (the package for the current file).
Note:Packages in the Java language itself begin with java. or javax.

QUESTION 57
Given:

```java
public class X implements Z
{
  public String toString()
  {
    return "I am X";
  }

  public static void main(String[] args)
  {
    Y myY = new Y();
    X myX = myY;
    Z myZ = myX;
    System.out.println(myZ);
  }
```

```
}

class Y extends X
{
   public String toString()
   {
      return "I am Y";
   }

}

interface Z { }
```

What is the reference type of myZ and what is the type of the object it references?

A. Reference type is Z; object type is Z.
B. Reference type is Y; object type is Y.
C. Reference type is Z; object type is Y.
D. Reference type is X; object type is Z.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation: Note: Because Java handles objects and arrays by reference, classes and array types are known as reference types.

QUESTION 58
Given:

```
class SampleClass
{

}

class AnotherSampleClass extends SampleClass
{

}
```

```
class Test
{
    public static void main(String[] args)
    {
        SampleClass sc = new SampleClass();
        AnotherSampleClass asc = new AnotherSampleClass();
        sc = asc;
        System.out.println("sc: " + sc.getClass());
        System.out.println("asc: " + asc.getClass());
    }
}
```

What is the result?

A. sc: class.Object
    asc: class.AnotherSampleClass

B. sc: class.SampleClass
    asc: class.AnotherSampleClass

C. sc: class.AnotherSampleClass
    asc: class.SampleClass

D. sc: class.AnotherSampleClass
    asc: class.AnotherSampleClass

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation: Note: The getClass method Returns the runtime class of an object. That Class object is the object that is locked by static synchronized methods of the represented class.

Note: Because Java handles objects and arrays by reference, classes and array types are known as reference types.

QUESTION 59
Given the code fragment:

```
public static void main(String[] args)
{
    String [] table = {"aa", "bb", "cc"};
    int ii = 0;
```

```
    for (String ss:table)
    {
        while (ii < table.length)
        {
            System.out.println (ii);
            ii++;
            break;
        }
    }
}
```

How many times is 2 printed?

A. zero

B. once

C. twice

D. thrice

E. it is not printed because compilation fails

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation: The outer loop will run three times, one time each for the elements in table. The break statement breaks the inner loop immediately each time. 2 will be printed once only.
Note: If the line int ii = 0; is missing the program would not compile.

QUESTION 60
Given:

```
public class SampleClass
{
    public static void main(String[] args)
    {
        SampleClass sc, scA, scB;
        sc = new SampleClass();
        scA = new SampleClassA();
        scB = new SampleClassB();
```

```
        System.out.println("Hash is : " + sc.getHash() + ", " + scA.getHash() + ", " + scB.getHash());
    }

    public int getHash()
    {
        return 111111;
    }
}

class SampleClassA extends SampleClass
{
    public long getHash()
    {
        return 44444444;
    }
}

class SampleClassB extends SampleClass
{
    public long getHash()
    {
        return 999999999;
    }
}
```

What is the result?

A. Compilation fails
B. An exception is thrown at runtime
C. There is no result because this is not correct way to determine the hash code "Pass Any Exam. Any Time." - www.actualtests.com 51
    Oracle 1z0-803 Exam
D. Hash is: 111111, 44444444, 999999999

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Explanation: The compilation fails as SampleClassA and SampleClassB cannot override SampleClass because the return type of SampleClass is int, while the
return type of SampleClassA and SampleClassB is long.
Note: If all three classes had the same return type the output would be:

Hash is : 111111, 44444444, 999999999

QUESTION 61
Which two will compile, and can be run successfully using the command:

Java fred1 hello walls

A. class Fred1{
    public static void main (String args) {
    System.out.println(args[1]);
    }
    }
B. class Fred1{
    public static void main (String [] args) {
    System.out.println(args[2]);
    }
    }
C. class Fred1 {
    public static void main (String [] args) {
    System.out.println (args);
    }
    }
D. class Fred1 {
    public static void main (String [] args) {
    System.out.println (args [1]);
    }
    }

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:
Explanation: Throws java.lang.ArrayIndexOutOfBoundsException: 2 at certquestions.Fred1.main(Fred1.java:3)
C. Prints out: [Ljava.lang.String;@39341183
D. Prints out: walls

QUESTION 62
Given:

public abstract class Wow {

```
    private int wow;

    public wow(int wow) {

        this.wow = wow;

    }

    public void wow() {}

    private void wowza() {}

}
```

What is true about the class Wow?

A. It compiles without error.
B. It does NOT compile because an abstract class CANNOT have private methods.
C. It does NOT compile because an abstract class CANNOT have instance variables.
D. It does NOT compile because an abstract class must have at least one abstract method.
E. It does NOT compile because an abstract class must have a constructor with no arguments.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation: An abstract class is a class that is declared abstract--it may or may not include abstract methods (not B, not D). Abstract classes cannot be instantiated, but they can be subclassed.
The code compiles with a failure for line 'public wow (int wow) {}

QUESTION 63
Given:

```
class X
{
    static void m(int i)
    {

    }
```

```
    public static void main(String[] args)
    {
        int j = 12;

        m (j);

        System.out.println(j);
    }
}
```

What is the result?

A. 7
B. 12
C. 19
D. Compilation fails
E. An exception is thrown at run time

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 64
Which two statements are true?

A. An abstract class can implement an interface.
B. An abstract class can be extended by an interface.
C. An interface CANNOT be extended by another interface.
D. An interface can be extended by an abstract class.
E. An abstract class can be extended by a concrete class.
F. An abstract class CANNOT be extended by an abstract class.

Correct Answer: AE
Section: (none)
Explanation

Explanation/Reference:
Explanation: Reference: http://docs.oracle.com/javase/tutorial/java/IandI/abstract.htm

QUESTION 65
Given:

```
class Overloading {

    int x(double d) {

        System.out.println("one");

        return 0;

    }

    String x(double d) {

        System.out.println("two");

        return null;

    }

    double x(double d) {

        System.out.println("three");

        return 0.0;

    }

    public static void main(String[] args) {

        new Overloading().x(4.0)

    }

}
```

What is the result?

A. One
B. Two
C. Three
D. Compilation fails

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation: overloading of the x method fails as the input argument in all three cases are double. To use overloading of methods the argument types must be different.

Note: The Java programming language supports overloading methods, and Java can distinguish between methods with different method signatures. This means that methods within a class can have the same name if they have different parameter lists

QUESTION 66
The catch clause argument is always of type_____.

A. Exception
B. Exception but NOT including RuntimeException
C. Throwable
D. RuntimeException
E. CheckedException
F. Error

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation: Because all exceptions in Java are the sub-class ofjava.lang.Exception class, you can have a single catch block that catches an exception of type Exception only. Hence the compiler is fooled into thinking that this block can handle any exception.
See the following example:

try
{
// ...

```
}
catch(Exception ex)
{
// Exception handling code for ANY exception
}
```

You can also use the java.lang.Throwable class here, since Throwable is the parent class for the application-specific Exception classes. However, this is discouraged in Java programming circles. This is because Throwable happens to also be the parent class for the non-application specific Error classes which are not meant to be handled explicitly as they are catered for by the JVM itself.

Note: The Throwable class is the superclass of all errors and exceptions in the Java language. Only objects that are instances of this class (or one of its subclasses) are thrown by the Java Virtual Machine or can be thrown by the Java throw statement. A throwable contains a snapshot of the execution stack of its thread at the time it was created. It can also contain a message string that gives more information about the error.

QUESTION 67
Given the code fragment:

1. ArrayList<Integer> list = new ArrayList<>(1);

2. list.add(1001);

3. list.add(1002);

4. System.out.println(list.get(list.size()));

What is the result?

A. Compilation fails due to an error on line 1.
B. An exception is thrown at run time due to error on line 3
C. An exception is thrown at run time due to error on line 4
D. 1002

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation: The code compiles fine.
At runtime an IndexOutOfBoundsException is thrown when the second list item is added.

QUESTION 68
View the Exhibit.

```java
public class Hat {

    public int ID = 0;

    public String name = "hat";

    public String size = "One Size Fit All";

    public String color = "";

    public String getName() {
        return name;
    }

    public void setName(String name) {

        this.name = name;

    }

}
```

Given

```java
public class TestHat {

    public static void main(String[] args) {

        Hat blackCowboyHat = new Hat();

    }

}
```

Which statement sets the name of the Hat instance?

A. blackCowboyHat.setName = "Cowboy Hat";
B. setName("Cowboy Hat");
C. Hat.setName("Cowboy Hat");

D. blackCowboyHat.setName("Cowboy Hat");

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:

QUESTION 69
Given:

```
public class Two
{
    public static void main(String[] args)
    {
        try
        {
            doStuff();
            system.out.println("1");
        }
        catch
        {
            system.out.println("2");
        }
    }

    public static void do Stuff()
    {
        if (Math.random() > 0.5) throw new RunTimeException(); doMoreStuff();
        System.out.println("3 ");
    }

    public static void doMoreStuff()
    {
        System.out.println("4");
    }
}
```

Which two are possible outputs?

A. 2

B. 4 3 1

C. 1 2 3

D. 1 3 4

Correct Answer: AB
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 70
Given:

```
public class MyFor
{
    public static void main(String[] args)
    {
        for (int ii = 0; ii < 4; ii++)
        {
            System.out.println("ii = "+ ii);
            ii = ii +1;
        }
    }
}
```

What is the result?

A. ii = 0
   ii = 2
B. ii = 0
   ii = 1
   ii = 2
   ii = 3
C. ii =
D. Compilation fails.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 71
Given the code fragment:

```
int [][] array2d = new int[2][3];

System.out.println("Loading the data.");

for ( int x = 0; x < array2d.length; x++)
{
   for ( int y = 0; y < array2d[0].length; y++)
   {
      System.out.println(" x = " + x);
      System.out.println(" y = " + y);

      // insert load statement here.
   }
}

System.out.println("Modify the data. ");

for ( int x = 0; x < array2d.length; x++)
{
   for ( int y = 0; y < array2d[0].length; y++)
   {
      System.out.println(" x = " + x);
      System.out.println(" y = " + y);

      // insert modify statement here.
   }
}
```

Which pair of load and modify statement should be inserted in the code? The load statement should set the array's x row and y column value to the sum of x and y

The modify statement should modify the array's x row and y column value by multiplying it by 2

A. Load statement: array2d(x,y) = x + y;
    Modify statement: array2d(x,y) = array2d(x,y) * 2
B. Load statement: array2d[x y] = x + y;

Modify statement: array2d[x y] = array2d[x y] * 2

C. Load statement: array2d[x,y] = x + y;
    Modify statement: array2d[x,y] = array2d[x,y] * 2

D. Load statement: array2d[x][y] = x + y;
    Modify statement: array2d[x][y] = array2d[x][y] * 2

E. Load statement: array2d[[x][y]] = x + y;
    Modify statement: array2d[[x][y]] = array2d[[x][y]] * 2

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 72
Given:

```
public class DoBreak1
{
    public static void main(String[] args)
    {
        String[] table = {"aa", "bb", "cc", "dd"};

        for (String ss: table)
        {
            if ( "bb".equals(ss))
            {
                continue;
            }

            System.out.println(ss);

            if ( "cc".equals(ss))
            {
                break;
            }
        }
    }
}
```

What is the result?

A. aa
   cc
B. aa
   bb
   cc
C. cc
   dd
D. cc
E. Compilation fails.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 73
1. class StaticMethods {

2. static void one() {

3. two();

4. StaticMethods.two();

5. three();

6. StaticMethods.four();

7. }

8. static void two() { }

9. void three() {

10. one();

11. StaticMethods.two();

12. four();

13. StaticMethods.four();

14. }

15. void four() { }

16. }

Which three lines are illegal?

A. line 3
B. line 4
C. line 5
D. line 6
E. line 10
F. line 11
G. line 12
H. line 13

Correct Answer: CDH
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 74
Which is a valid abstract class?

A. public abstract class Car {
      protected void accelerate();
      }
B. public interface Car {
      protected abstract void accelerate();
      }
C. public abstract class Car {

```
        protected final void accelerate();
        }
D.  public abstract class Car {
        protected abstract void accelerate();
        }
E.  public abstract class Car {
        protected abstract void accelerate() {
        //more car can do
        }}
```

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 75
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 ();
     bob.name = "Bob";
     bob.age = 18;
     bob.year = 1982;
   }
}

What is the result?

A. year is set to 1982.
B. bob.year is set to 1982
C. A runtime error is generated.
D. A compile time error is generated.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 76
Given the code fragment:

String name = "Spot";

int age = 4;

String str ="My dog " + name + " is " + age;

System.out.println(str);

And

StringBuilder sb = new StringBuilder();

Using StringBuilder, which code fragment is the best potion to build and print the following string My dog Spot is 4

A. sb.append("My dog " + name + " is " + age);
   System.out.println(sb);
B. sb.insert("My dog ").append(name + " is " + age);
   System.out.println(sb);
C. sb.insert("My dog ").insert(name).insert(" is ").insert(age);
   System.out.println(sb);
D. sb.append("My dog ").append(name).append(" is ").append(age);
   System.out.println(sb);

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 77
Given:

public class Main {

    public static void main(String[] args) {

        try {

            doSomething();

        } catch (SpecialException e) {

            System.out.println(e);

        }
    }

    static void doSomething() {

        int[] ages = new int[4];

```
        ages[4] = 17;

        doSomethingElse();

    }

    static void doSomethingElse() {

        throw new SpecialException("Thrown at end of doSomething() method");
    }

}
```

What is the output?

A. SpecialException: Thrown at end of doSomething() method
B. Error in thread "main" java.lang.
    ArrayIndexOutOfBoundseror
C. Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 4
    at Main.doSomething(Main.java:12)
    at Main.main(Main.java:4)
D. SpecialException: Thrown at end of doSomething() method at Main.doSomethingElse(Main.java:16)
    at Main.doSomething(Main.java:13)
    at Main.main(Main.java:4)

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation: The following line causes a runtime exception (as the index is out of bounds): ages[4] = 17;

A runtime exception is thrown as an ArrayIndexOutOfBoundsException.

Note: The third kind of exception (compared to checked exceptions and errors) is the runtime exception.
These are exceptional conditions that are internal to the application, and that the application usually cannot anticipate or recover from. These usually indicate
programming bugs, such as logic errors or improper use of an API.
Runtime exceptions are not subject to the Catch or Specify Requirement. Runtime exceptions are those indicated by RuntimeException and its subclasses.

QUESTION 78
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;

    }
}
```

Which line of code initializes a student instance?

A. Student student1;
B. Student student1 = Student.new();
C. Student student1 = new Student();
D. Student student1 = Student();

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 79
int [] array = {1,2,3,4,5};

for (int i: array) {

if (i < 2) {

keyword1 ;

}

System.out.println(i);

if (i == 3) {

keyword2 ;

}}

What should keyword1 and keyword2 be respectively, in oreder to produce output 2345?

A. continue, break
B. break, break
C. break, continue
D. continue, continue

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 80
int i, j=0;

i = (3* 2 + 4 +5) ;

j = (3 * ((2 + 4) + 5));

System.out.println("i:"+ i + "\nj": + j);

What is the result?

A. i:16

j:16
B. 16
C. i:15
j:33
D. 33
E. i:16
j:33
F. 15
G. i:15
j:16
H. 23

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation: It is obvious!

QUESTION 81
Given:

boolean log3 = (5.0 != 6.0) && (4 != 5);

boolean log4 = (4 != 4) || (4 == 4);

System.out.println("log3:"+ log3 + \nlog4" + log4);

What is the result?

A. log3:false
log4:true
B. log3:true
log4:true
C. log3:true
log4:false
D. log3:false
log4:false

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 82
Which statement will emoty 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:
No explanation yet!

QUESTION 83
Given:

Class StaticField {

   static int i = 7;

   public static void main(String[] args) {

      StaticFied obj = new StaticField();

      obj.i++;

      StaticField.i++;

      obj.i++;

      System.out.println(StaticField.i + " " + obj.i);

}

}

What is the result?

A. 10 10
B. 8 9
C. 9 8
D. 7 10

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 84
Which two are valid array declaration?

A. Object array[];
B. Boolean array[3];
C. int[] array;
D. Float[2] array;

Correct Answer: AC
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 85
Given:

class Overloading {

    int x(double d) {

```java
        System.out.println("one");

        return 0;

    }

    String x(double d) {

        System.out.println("two");

        return null;

    }

    double x(double d) {

        System.out.println("three");

        return 0.0;

    }

    public static void main(String[] args) {

        new Overloading().x(4.0);

    }

}
```

What is the result?

A. one
B. two
C. three
D. Compilation fails.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 86
Given:

```
public class MainMethod {

    void main() {

        System.out.println("one");

    }

    static void main(String args) {

        System.out.println("two");

    }

    public static void main(String[] args) {

        System.out.println("three");

    }

    void main(Object[] args) {

        System.out.println("four");

    }

}
```

What is printed out when the program is excuted?

A. one
B. two
C. three
D. four

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 87
Given:

```java
public class ScopeTest {

    int j, int k;

    public static void main(String[] args) {

        new ScopeTest().doStuff();
    }

    void doStuff() {

        int x = 5;

        doStuff2();

        System.out.println("x");

    }

    void doStuff2() {

        int y = 7;

        System.out.println("y");

        for (int z = 0; z < 5; z++) {

            System.out.println("z");

            System.out.println("y");
```

}
    }

Which two items are fields?

A. j
B. k
C. x
D. y
E. z

Correct Answer: AB
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 88
Given:

public class ForTest {

    public static void main(String[] args) {

        int[] arrar = {
            1, 2, 3
        };

        for (foo) {

        }

    }

}
Which three are valid replacements for foo so that the program will compiled and run?

A. int i: array

B. int i = 0; i < 1; i++

C. ;;

D. ; i < 1; i++

E. ; i < 1;

Correct Answer: ABC
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 89
Given the code fragment:

int b = 3;

if (!(b > 3)) {

   System.out.println("square ");

} {

   System.out.println("circle ");

}

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

What is the result?

A. square...

B. circle...

C. squarecircle...

D. Compilation fails.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 90
What is the proper way to defined a method that take two int values and returns their sum as an int value?

A. int sum(int first, int second) { first + second; }
B. int sum(int first, second) { return first + second; }
C. sum(int first, int second) { return first + second; }
D. int sum(int first, int second) { return first + second; }
E. void sum (int first, int second) { return first + second; }

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 91
Which two are Java Exception classes?

A. SecurityException
B. DuplicatePathException
C. IllegalArgumentException
D. TooManyArgumentsException

Correct Answer: AC
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 92
Given the for loop construct:

for (expr1 ; expr2 ; expr3)
{

statement;

}

Which two statements are true?

A. This is not the only valid for loop construct; there exits another form of for loop constructor.
B. The expression expr1 is optional. it initializes the loop and is evaluated once, as the loop begin.
C. When expr2 evaluates to false, the loop terminates. It is evaluated only after each iteration through the loop.
D. The expression expr3 must be present. It is evaluated after each iteration through the loop.

Correct Answer: BC
Section: (none)
Explanation

Explanation/Reference:
Explanation:
The for statement have this forms:
for (init-stmt; condition; next-stmt) {
body
}
There are three clauses in the for statement.
The init-stmt statement is done before the loop is started, usually to initialize an iteration variable. The condition expression is tested before each time the loop is done. The loop isn't executed if the boolean expression is false (the same as the while loop). The next-stmt statement is done after the body is executed. It typically increments an iteration variable.

QUESTION 93
Given:

public class StringReplace {

    public static void main(String[] args) {

        String message = "Hi everyone!";

        System.out.println("message = " + message.replace("e", "X"));
    }

}

What is the result?

A. message = Hi everyone!
B. message = Hi XvXryonX!
C. A compile time error is produced.
D. A runtime error is produced.
E. message =
F. message = Hi Xveryone!

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 94
Which two statements are true for a two-dimensional array?

A. It is implemented as an array of the specified element type.
B. Using a row by column convention, each row of a two-dimensional array must be of the same size
C. At declaration time, the number of elements of the array in each dimension must be specified
D. All methods of the class Object may be invoked on the two-dimensional array.

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 95
Which three statements are benefits of encapsulation?

A. allows a class implementation to change without changing the clients
B. protects confidential data from leaking out of the objects
C. prevents code from causing exceptions
D. enables the class implementation to protect its invariants

E. permits classes to be combined into the same package

F. enables multiple instances of the same class to be created safely

Correct Answer: ABD
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

Exam B

QUESTION 1
Which code fragment cause 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;

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Type mismatch: cannot convert from double to float

QUESTION 2
Given the code fragment:

```
String[] colors = {"red", "blue", "green", "yellow", "maroon", "cyan"};
```

Which code fragment prints blue, cyan, ?

```
A. for (String c: colors) {
       if (c.length() != 4) {
           continue;
       }
       System.out.print(c + ", ");
   }
B. for (String c: colors[]) {
       if (c.length() <= 4) {
           continue;
       }
       System.out.print(c + ", ");
   }
```

```
C. for (String c: String[] colors) {
       if (c.length() >= 3) {
           continue;
       }
       System.out.print(c + ", ");
   }
D. for (String c: colors) {
       if (c.length() != 4) {
           System.out.print(c + ", ");
           continue;
       }
   }
```

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 3
Given:

```
class X {
    static void m(int[] i) {
        i[0] += 7;
    }
    public static void main(String[] args) {
        int[] j = new int[1];
        j[0] = 12;
        m(j);
        System.out.println(j[0]);
    }
}
```

What is the result?

A. 7
B. 12
C. 19
D. Compilation fails.

E. An exception is thrown at runtime.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 4
Given:

```
class MarksOutOfBoundsException extends IndexOutOfBoundsException {}
public class GradingProcess {
    void verify(int marks) throws IndexOutOfBoundsException {
        if (marks > 100) {
            throw new MarksOutOfBoundsException();
        }
        if (marks > 50) {
            System.out.print("Pass");
        } else {
            System.out.print("Fail");
        }
    }
    public static void main(String[] args) {
        int marks = Integer.parseInt(args[2]);
        try {
            new GradingProcess().verify(marks);
        } catch (Exception e) {
            System.out.print(e.getClass());
        }
    }
}
```

And the command line invocation:
```
java GradingProcess 89 50 104
```

What is the result?

A. Pass
B. Fail
C. class MarksOutOfBoundsException

D. class IndexOutOfBoundsException

E. class Exception

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 5
Given:
```
1. interface Pet { }
2. class Dog implements Pet { }
3. class Beagle extends Dog { }
```

Which three are valid?

A. Pet a = new Dog();

B. Pet b = new Pet();

C. Dog f = new Pet();

D. Dog d = new Beagle();

E. Pet e = new Beagle();

F. Beagle c = new Dog();

Correct Answer: ADE
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 6
Given the code fragment:

```
StringBuilder sb = new StringBuilder();
sb.append("World");
```

Which fragment prints Hello World?

A. ```
   sb.insert(0, "Hello ");
   System.out.println(sb);
   ```
B. ```
   sb.append(0, "Hello ");
   System.out.println(sb);
   ```
C. ```
   sb.add(0, "Hello ");
   System.out.println(sb);
   ```
D. ```
   sb.set(0, "Hello ");
   System.out.println(sb);
   ```

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 7
Given:

```
package pkg1;
class Bb {}
public class Ee {
    private Ee() {}
}
package pkg2;
final class Ww;
package pkg3;
public abstract class Dd {
```

```
    void m() {}
}

And,
1. package pkg4;
2. import pkg1.*;
3. import pkg2.*;
4. import pkg3.*;
5. // insert a class definition here
```

Which two class definitions, when inserted independently at line 5, enable the code to compile?

A. class Cc extends Bb { }

B. class Cc extends Ww { }

C. class Cc extends Ee { }

D. class Cc extends Dd { }

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 8
Given the code fragment:
```
class Student {
String name;
int age;
}
And,
1. public class Test {
2. public static void main (String[] args) {
3. Student s1 = new Student();
4. Student s2 = new Student();
5. Student s3 = new Student();
6. s1 = s3;
7. s3 = s2;
8. s2 = null;
9. }
10. }
```

Which statement is true?

A. After line 8, three objects are eligible for garbage collection.
B. After line 8, two objects are eligible for garbage collection.
C. After line 8, one object is eligible for garbage collection.
D. After line 8, none of the objects are eligible for garbage collection.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 9
Given:

```
public class Test {
    public static void main(String[] args) {
        char[] arr = {
            97, '\t', 'e', '\n', 'i', '\t', 'o'
        };
        for (char
        var: arr) {
            System.out.print(var);
        }
        System.out.print("\nThe length is :" + arr.length);
    }
}
```

What is the result?

A. a e
    The length is : 2
B. a e
    i o
    The length is : 4
C. aeio
    The length is : 4
D. a e
    i o

The length is : 7
E. Compilation fails.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 10
Given the class definitrions:

```
class Shape { }
class Square extends Shape { }
```
Given the variable declarations:
```
Shape shape1 = null;
Square square1 = null;
```

Which four compile?

A. shape1 = (Square) new Square();

B. shape1 = new Square();

C. square1 = (Square) new Shape();

D. square1 = new Shape();

E. square1 = new Square();
    shape1 = square1;

F. shape1 = new Shape();
    square1 = shape1;

Correct Answer: ABCE
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 11
Given the code fragments:

```
9. class Student {
10. int rollnumber;
11. String name;
12. List courses = new ArrayList();
13. // insert code fragment here
14. public String toString() {
15. return rollnumber + " : " + name + " : " + courses;
16. }
17. }
```

And,

```
public class Test {
    public static void main(String[] args) {
        List cs = new ArrayList();
        cs.add("Java");
        cs.add("C");
        Student s = new Student(123, "Fred", cs);
        System.out.println(s);
    }
}
```

Which code fragment, when inserted at line 13, enables class Test to print 123 : Fred : [Java, C] ?

A. ```
private Student(int i, String name, List cs) {
   /* initialization code goes here */
   }
```
B. ```
public void Student(int i, String name, List cs) {
   /* initialization code goes here */
   }
```
C. ```
Student(int i, String name, List cs) {
   /* initialization code goes here */
   }
```
D. ```
Student(int i, String name, ArrayList cs) {
   /* initialization code goes here */
   }
```

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 12
Given:

```
public class Test2 {
    public static void main(String[] args) {
        int ar1[] = {
            2, 4, 6, 8
        };
        int ar2[] = {
            1, 3, 5, 7, 9
        };
        ar2 = ar1;
        for (int e2: ar2) {
            System.out.print(" " + e2);
        }
    }
}
```

What is the result?

A. 2 4 6 8

B. 2 4 6 8 9

C. 1 3 5 7

D. 1 3 5 7 9

E. Compilation fails

F. An exception is thrown at runtime

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 13
Given:

```
public class Test2 {
    public static void doChange(int[] arr) {
        for (int pos = 0; pos < arr.length; pos++) {
```

```
            arr[pos] = arr[pos] + 1;
        }
    }
    public static void main(String[] args) {
        int[] arr = {
            10, 20, 30
        };
        doChange(arr);
        for (int x: arr) {
            System.out.print(x + ", ");
        }
        doChange(arr);
        System.out.print(arr[0] + ", " + arr[1] + ", " + arr[2]);
    }
}
```

What is the result?

A. 11, 21, 31, 11, 21, 31
B. 11, 21, 31, 12, 22, 32
C. 12, 22, 32, 12, 22, 32
D. 10, 20, 30, 10, 20, 30

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 14
Given:

```
public class Natural {
    private int i;
    void disp() {
        while (i <= 5) {
            for (int i = 1; i <= 5;) {
                System.out.print(i + " ");
                i++;
            }
            i++;
```

```
        }
    }
    public static void main(String args[]) {
        new Natural().disp();
    }
}
```

What is the result?

A. Prints 1 2 3 4 5 once
B. Prints 1 3 5 once
C. Prints 1 2 3 4 5 five times
D. Prints 1 2 3 4 5 six times
E. Compilation fails

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 15
Given:

```
public class CheckIt {
    public static void main(String[] args) {
        if (doCheck()) {
            System.out.print("square ");
        }
        System.out.print("...");
    }
    public static int doCheck() {
        return 0;
    }
}
```

What is the result?

A. square...
B. ...

C. Compilation fails.

D. An exception is thrown at runtime.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 16
Given:

```
public class Test { }
```

From which class does the Java compiler implicitly derive Test?

A. Object

B. Class

C. An anonymous class

D. Objects

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 17
Given:

```
class Test {
    public static void main(String[] args) {
        int day = 1;
        switch (day) {
            case "7":
                System.out.print("Uranus");
            case "6":
                System.out.print("Saturn");
            case "1":
```

```
            System.out.print("Mercury");
        case "2":
            System.out.print("Venus");
        case "3":
            System.out.print("Earth");
        case "4":
            System.out.print("Mars");
        case "5":
            System.out.print("Jupiter");
        }
    }
}
```

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

A. adding a break statement after each print statement

B. adding a default section within the switch code-block

C. changing the string literals in each case label to integer

D. changing the type of the variable day to String

E. arranging the case labels in ascending order

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 18
Given:

```
class App {
    private boolean flag;
    public void displayResult() {

        int result = flag ? 5 : 10;
        System.out.print("Result = " + result++);
    }
    public static void main(String[] args) {
        new App().displayResult();
    }
}
```

What is the result?

A. Compilation fails.

B. Result = 5

C. Result = 6

D. Result = 10

E. Result = 11

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 19
Given:

```
abstract class A1 {
    public abstract void m1();
    public void m2() {
        System.out.println("Green");
    }
}
abstract class A2 extends A1 {
    public abstract void m3();
    public void m1() {
        System.out.println("Cyan");
    }
    public void m2() {
        System.out.println("Blue");
    }
}
public class A3 extends A2 {
    public void m1() {
        System.out.println("Yellow");
    }
    public void m2() {
        System.out.println("Pink");
    }
    public void m3() {
        System.out.println("Red");
```

```
        }
    public static void main(String[] args) {
        A2 tp = new A3();
        tp.m1();
        tp.m2();
        tp.m3();
    }
}
```

What is the result?

A. Yellow
   Pink
   Red
B. Cyan
   Blue
   Red
C. Cyan
   Green
   Red
D. Compilation fails

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 20
Given:

```
class X {
    public void mX() {
        System.out.println("Xm1");
    }
}
class Y extends X {
    public void mX() {
        System.out.println("Xm2");
    }
    public void mY() {
```

```
            System.out.println("Ym");
        }
}
public class Test {
    public static void main(String[] args) {
        X xRef = new Y();
        Y yRef = (Y) xRef;
        yRef.mY();
        xRef.mX();
    }
}
```

What is the result?

A. Ym
   Xm2
B. Ym
   Xm1
C. Compilation fails.
D. Xm1
   Ym

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 21
Given:

```
class Prime {
    int num;
    Prime(int num) {
        this.num = num;
    }
}
public class Test {
    public static void main(String[] args) {
        Prime obj1 = new Prime(13);
        Prime obj2 = new Prime(13);
```

```
        if (obj1 == obj2) {
            System.out.println("Equal");
        } else {
            System.out.println("Not equal");
        }
        if (obj1.equals(obj2)) {
            System.out.println("Equal");
        } else {
            System.out.println("Not equal");
        }
    }
}
```

What is the result?

A. Equal
   Not equal
B. Not equal
   Equal
C. Equal
   Equal
D. Not equal
   Not equal

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 22
Given the code fragment:

```
String[] cartoons = {
    "tom", "jerry", "micky", "tom"
};
int counter = 0;
And,
if ("tom".equals(cartoons[0])) {
    counter++;
} else if ("tom".equals(cartoons[1])) {
```

```
      counter++;
} else if ("tom".equals(cartoons[2])) {
      counter++;
} else if ("tom".equals(cartoons[3])) {
      counter++;
}
System.out.print(counter);
```

What is the result?

A. 1

B. 2

C. 4

D. 0

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 23
Given:

```
class X {
    static void m(StringBuilder sb1) {
       sb1.append("er");
    }
    public static void main(String[] args) {
       StringBuilder sb2 = new StringBuilder("moth");
       m(sb2);
       System.out.println(sb2);
    }
}
```

What is the result?

A. moth

B. er

C. mother

D. Compilation fails.

E. An exception is thrown at run time

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 24
Given:

```
public class DoWhile1 {
    public static void main(String[] args) {
        int i = 2;
        do {
            System.out.println(i);
        } while (--i);
    }
}
```

What is the result?

A. 1

B. 2

C. An exception is thrown at runtime

D. The loop executes infinite times

E. Compilation fails

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Type mismatch: cannot convert from int to boolean

QUESTION 25
Given:

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

Which change will make this code compile?

A. Add the public modifier to the declaration of class X

B. Remove the private modifier from the X() constructor

C. Add the protected modifier to the declaration of the one() method

D. Remove the Y() constructor

E. Remove the private modifier from Y() constructor

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 26
Given the code fragment:

```
class Test2 {
    int fvar;
    static int cvar;
    public static void main(String[] args) {
        Test2 t = new Test2();
        // insert code here to write field variables
    }
}
```

Which two code fragments, inserted independently, enable the code to compile?

A. t.fvar = 200;

B. cvar = 400;

C. fvar = 200;
   cvar = 400;

D. this.fvar = 200;
   this.cvar = 400;

E. t.fvar = 200;
   Test2.cvar = 400;

F. this.fvar = 200;
   Test2.cvar = 400;

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 27
Given:

```
public class App {
    public static void main(String[] args) {
        char[] arr = {
            'A', 'e', 'I', 'O', 'u'
        };
        int count = 0;
        for (int i = 0; i < arr.length; i++) {
            switch (arr[i]) {
                case 'A':
                    continue;
                case 'a':
                    count++;
                    break;
                case 'E':
                    count++;
                    break;
                case 'I':
                    count++;
                    continue;
                case 'O':
```

```
            count++;
            break;
        case 'U':
            count++;
    }
}
System.out.print("Total match found: " + count);
    }
}
```

What is the result?

A. Total match found: 1
B. Total match found: 2
C. Total match found: 3
D. Total match found: 5

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 28
Given:

```
public class Series {
    private boolean flag;
    public void displaySeries() {
        int num = 2;
        while (flag) {
            if (num % 7 == 0) flag = false;
            System.out.print(num);
            num += 2;
        }
    }
    public static void main(String[] args) {
        new Series().displaySeries();
    }
}
```

What is the result?

A. 2 4 6 8 10 12
B. 2 4 6 8 10 12 14
C. Compilation fails
D. The program prints multiple of 2 infinite times
E. The program prints nothing

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 29
Given:

```
public class MarkList {
    int num;
    public static void graceMarks(MarkList obj4) {
        obj4.num += 10;
    }
    public static void main(String[] args) {
        MarkList obj1 = new MarkList();
        MarkList obj2 = obj1;
        MarkList obj3 = null;
        obj2.num = 60;
        graceMarks(obj2);
    }
}
```

How many objects are created in the memory at runtime?

A. 1
B. 2
C. 3
D. 4

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 30
Given:

```
class X {
    String str = "default";
    int ivalue = 17;
    X(String s) {
        str = s;
    }
    X(int i) {
        ivalue = i;
    }
    void print() {
        System.out.println(str + " " + ivalue);
    }
    public static void main(String[] args) {
        new X("hello").print();
        new X(92).print();
    }
}
```

What is the result?

A. default 17
    hello 92
B. hello 92
    default 17
C. hello 17
    default 92
D. default 92
    hello 17

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 31
Given:

```
abstract class X {
    public abstract void methodX();
}
interface Y {
    public void methodY();
}
```

Which two code fragments are valid?

A. ```
class Z extends X implements Y {
public void methodZ() { }
}
```
B. ```
abstract class Z extends X implements Y {
public void methodZ() { }
}
```
C. ```
class Z extends X implements Y {
public void methodX() { }
}
```
D. ```
abstract class Z extends X implements Y {
}
```
E. ```
class Z extends X implements Y {
public void methodY() { }
}
```

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

**Explanation/Reference:**
No explanation yet!

**QUESTION 32**
Which four statements are true regarding exception handling in Java?

A. In multicatch blocks, the subclass catch handler must be caught after the superclass catch handler.

B. A try block must be followed by either a catch or finally block

C. The Exception class is the superclass of all errors and exception in the Java language

D. A single catch block can handle more than one type of exception

E. A checked exception must be caught explicitly

F. In a catch block than can handle more than one exception, the subclass exception handler must be caught before the superclass exception handler

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

**Explanation/Reference:**
No explanation yet!

**QUESTION 33**
Given the code fragment:

```
class MySearch {
    public static void main(String[] args) {
        String url = "http://www.domain.com/index.html";
        if (XXXX) {
            System.out.println("Found");
        }
    }
}
```

Which code fragment, replace with XXXX, enables the code to print Found?

A. url.indexOf("com") != -1

B. url.indexOf("com")

C. url.indexOf("com") == 19

D. url.indexOf("com") != false

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 34
Which three statements are true about the structure of a Java class?

A. A class can have only one private constructor.
B. A method can have the same name as a field.
C. A class can have overloaded static methods.
D. A public class must have a main method.
E. The methods are mandatory components of a class.
F. The fields need not be initialized before use.

Correct Answer: BCF
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 35
Given the code fragment:

```
class Lang {
    private String category = "procedura1";
    public static void main(String[] args) {
        Lang obj1 = new Lang();
        Lang obj2 = new Lang();
        if (obj1.category == obj2.category) {
            System.out.println("Equal");
        } else {
            System.out.println("Not equal");
        }
        if (obj1.category.equals(obj2.category)) {
            System.out.println("Equal");
        } else {
            System.out.println("Not equal");
        }
    }
}
```

What is the result?

A. Equal

Not equal
B. Not equal
Equal
C. Equal
Equal
D. Not equal
Not equal

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 36
Given the code fragment:

```
public static void main(Stirng[] args) {
    int x = 353;
    int j = x++;
    switch (j) {
        case 317:
        case 353:
        case 367:
            System.out.println("Is a prime number.");
        case 353:
        case 363:
            System.out.println("Is a palindrome.");
            break;
        default:
            System.out.println("Invalid value.");
    }
}
```

What is the result?

A. Compilation fails
B. Is a prime number.
C. Is a palindrome.
D. Is a prime number.

Is a palindrome.

E. Invalid value.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
Duplicate case
Duplicate case

QUESTION 37
Given the code fragment:

```
boolean log1 = (1 < 9) && (1 > 5);
boolean log2 = (3 == 4) || (3 == 3);
System.out.println("log1:" + log1 + "\nlog2:" + log2);
```

What is the result?

A. log1: false
   log2: true

B. log1: true
   log2: true

C. log1: false
   log2: false

D. log1: true
   log2: false

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 38
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. cbo

C. bo

D. nb

E. An exception is thrown at runtime

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 6 at java.lang.String.substring(Unknown Source)

QUESTION 39
Given:

```
class Cake {
    int model;
    String flavor;
    Cake() {
        model = 0;
        flavor = "Unknown";
    }
}
public class Test {
    public static void main(String[] args) {
        Cake c = new Cake();
        bake1(c);
        System.out.println(c.model + " " + c.flavor);
        bake2(c);
        System.out.println(c.model + " " + c.flavor);
```

```
    }
    public static Cake bake1(Cake c) {
        c.flavor = "Strawberry";
        c.model = 1200;
        return c;
    }
    public static void bake2(Cake c) {
        c.flavor = "Chocolate";
        c.model = 1230;
        return;
    }
}
```

What is the result?

A. 0 Unknown
    0 Unknown

B. 1200 Strawberry
    1200 Strawberry

C. 1200 Strawberry
    1230 Chocolate

D. Compilation fails

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 40
Given:

```
class X {
    static void m(String s1) {
        s1 = "acting";
    }
    public static void main(String[] args) {
        String s2 = "action";
        m(s2);
        System.out.println(s2);
    }
```

}

What is the result?

A. acting
B. action
C. Compilation fails
D. An exception is thrown at runtime

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 41
Given:

```
public class Painting {
    private String type;
    public String getType() {
        return type;
    }
    public void setType(String type) {
        this.type = type;
    }
    public static void main(String[] args) {
        Painting obj1 = new Painting();
        Painting obj2 = new Painting();
        obj1.setType(null);
        obj2.setType("Fresco");
        System.out.print(obj1.getType() + " : " + obj2.getType());
    }
}
```

What is the result?

A. : Fresco
B. null : Fresco

C. Fresco : Fresco

D. A NullPointerException is thrown at runtime

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 42
Which two statement correctly describe checked exception?

A. These are exceptional conditions that a well-written application should anticipate and recover from.

B. These are exceptional conditions that are external to the application, and that the application usually cannot anticipate or recover from.

C. These are exceptional conditions that are internal to the application, and that the application usually cannot anticipate or recover from.

D. Every class that is a subclass of RuntimeException and Error is categorized as checked exception.

E. Every class that is a subclass of Exception, excluding RuntimeException and its subclasses, is categorized as checked exception.

Correct Answer: AE
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 43
Given the code fragment:

```
System.out.println( 28 + 5 <= 4 + 29 );
System.out.println( ( 28 + 5 ) <= ( 4 + 29) );
```

What is the result?

A. 28false29
   true

B. 285 < 429
   true

C. true

true
D. Compilation fails

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 44
Given:

```
public class Access {
    private int x = 0;
    private int y = 0;
    public static void main(String[] args) {
        Access accApp = new Access();
        accApp.printThis(1, 2);
        accApp.printThat(3, 4);
    }
    public void printThis(int x, int y) {
        x = x;
        y = y;
        System.out.println("x:" + this.x + " y:" + this.y);
    }
    public void printThat(int x, int y) {
        this.x = x;
        this.y = y;
        System.out.println("x:" + this.x + " y:" + this.y);
    }
}
```

What is the result?

A. x:1 y:2
    x:3 y:4
B. x:0 y:0
    x:3 y:4
C. x:3 y:4
    x:0 y:0

D. x:3 y:4
   x:1 y:2

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 45
Given:

```
public class Calculator {
    public static void main(String[] args) {
        int num = 5;
        sum;
        do {
            sum += sum;
        } while ((num--) > 1);
        System.out.println("The sum is " + sum + ".");
    }
}
```

What is the result?

A. The sum is 2.
B. The sum is 14.
C. The sum is 15.
D. The loop executes infinite times.
E. Compilation fails.

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
The local variable sum may not have been initialized
The local variable sum may not have been initialized

QUESTION 46
Given the code fragment:

```
5. // insert code here
6.
7. arr[0] = new int[3];
8. arr[0][0] = 1;
9. arr[0][1] = 2;
10. arr[0][2] = 3;
11.
12. arr[1] = new int[4];
13. arr[1][0] = 10;
14. arr[1][1] = 20;
15. arr[1][2] = 30;
16. arr[1][3] = 40;
```

Which two statements, when inserted independently at line 5, enable the code to compile?

A. int [][] arr = null;

B. int [][] arr = new int[2];

C. int [][] arr = new int[2][];

D. int [][] arr = new int[][4];

E. int [][] arr = new int[2][0];

F. int [][] arr = new int[0][4];

Correct Answer: CE
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 47
Given the code fragment:

```
12. for (int row = 4; row > 0; row--) {
13. int col = row;
14. while (col <= 4) {
15. System.out.print(col);
16. col++;
```

```
17. }
18. System.out.println();
19. }
```

What is the result?

A. 4
    34
    234
    1234

B. 4
    43
    432
    4321

C. 4321
    432
    43
    4

D. 4567
    345
    23
    1

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 48
Given:

```
public class Test3 {
    public static void main(String[] args) {
        double[] array = {
            10, 20.23, 'c', 300.00f
        };
        for (double d: array) {
            d = d + 10;
            System.out.print(d + " ");
        }
```

```
        }
}
```

What is the result?

A. 20.0 30.23 109.0 310.0
B. 20.0 30.23 c10 310.0
C. Compilation fails.
D. An exception is thrown at runtime.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 49
Given:

```
public class Test {
    int sum = 0;
    public void doCheck(int number) {
        if (number % 2 == 0) {
            break;
        } else {
            for (int i = 0; i < number; i++) {
                sum += i;
            }
        }
    }
    public static void main(String[] args) {
        Test obj = new Test();
        System.out.print("Red " + obj.sum);
        obj.doCheck(2);
        System.out.print("Orange " + obj.sum);
        obj.doCheck(3);
        System.out.print("Green " + obj.sum);
    }
}
```

What is the result?

A. Red 0
   Orange 0
   Green 3
B. Red 0
   Orange 0
   Green 6
C. Red 0
   Orange 1
   Green 4
D. Compilation fails

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 50
Given this code in a file Traveler.java:

```
class Tours {
    public static void main(String[] args) {
        System.out.print("Happy Journey! " + args[1]);
    }
}
public class Traveler {
    public static void main(String[] args) {
        Tours.main(args);
    }
}
```

And the commands:

```
javac Traveler.java
java Traveler Java Duke
```

What is the result?

A. Happy Journey! Duke

B. Happy Journey! Java
C. An exception is thrown at runtime.
D. The program fails to execute due to a runtime error.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 51
Which statement is true about the default constructor of a top-level class?

A. It can take arguments.
B. It has private access modifier in its declaration.
C. It can be overloaded.
D. The default constructor of a subclass always invokes the no-argument constructor of its superclass.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 52
Given:

```
public class X implements Z {
    public String toString() {
        return "X ";
    }
    public static void main(String[] args) {
        Y myY = new Y();
        X myX = myY;
        Z myZ = myX;
        System.out.print(myX);
        System.out.print((Y) myX);
        System.out.print(myZ);
    }
```

```
}
class Y extends X {
    public String toString() {
        return "Y ";
    }
}
interface Z {}
```

What is the result?

A. X X X

B. X Y X

C. Y Y X

D. Y Y Y

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 53
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);
```

```
        }
}
```

What is the result?

A. a, e
   i, o

B. a, e
   o, o

C. e, e
   i, o

D. e, e
   o, o

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 54
Given:

```
class Star {
    public void doStuff() {
        System.out.println("Twinkling Star");
    }
}
interface Universe {
    public void doStuff();
}
class Sun extends Star implements Universe {
    public void doStuff() {
        System.out.println("Shining Sun");
    }
}
public class Bob {
    public static void main(String[] args) {
        Sun obj2 = new Sun();
        Star obj3 = obj2;
        ((Sun) obj3).doStuff();
```

```
            ((Star) obj2).doStuff();
            ((Universe) obj2).doStuff();
        }
}
```

What is the result?

A. Shining Sun
    Shining Sun
    Shining Sun

B. Shining Sun
    Twinkling Star
    Shining Sun

C. Compilation fails.

D. A ClassCastException is thrown at runtime.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 55
Given:

```
public class TestA extends Root {
    public static void main(String[] args) {
        Root r = new TestA();
        System.out.println(r.method1()); // line n1
        System.out.println(r.method2()); // line n2
    }
}
class Root {
    private static final int MAX = 20000;
    private int method1() {
        int a = 100 + MAX; // line n3
        return a;
    }
    protected int method2() {
        int a = 200 + MAX; // line n4
        return a;
```

```
        }
}
```

Which line cause a compilation error?

A. line n1
B. line n2
C. line n3
D. line n4

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 56
Given:

```
public class VarScope {
    public static void main(Stirng[] args) {
        String color = "red";
        int qty = 10;
        if (color.equals("red")) { // line n1
            int amount = qty * 10;
        } else if (color.equals("green")) {
            int amount = qty * 15; // line n2
        } else if (color.equals("blue")) {
            int amount = qty * 5; // line n3
        }
        System.out.print(amount); // line n4
    }
}
```

Which line causes a compilation error?

A. line n1
B. line n2
C. line n3
D. line n4

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 57
Given:

```
public class Test {
    public static void main(String[] args) {
        int i = 1;
        do {
            if (i % 2 == 0) continue;
            if (i == 5) break;
            System.out.print(i + "\t");
            i++;
        } while (true);
    }
}
```

What is the result?

A. 1 3
B. 2 4
C. The program prints nothing.
D. Prints 1 infinite times.
E. Prints 1 once and the loop executes infinite times.

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 58
Which three statements are true regarding exception handling in Java?

A. A try block can be followed by multiple finally blocks.

B. A try block can be followed by a catch or finally block.

C. In multiple catch blocks, the superclass catch handler must be caught after the subclass catch handler

D. An unchecked exception must be caught explicitly.

E. A finally block can be written before the catch block.

F. Any Exception subclass can be used as the parent class of a user-defined exception.

Correct Answer: BCF
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 59
Given:

```
public class Test {
    public static void main(String[] args) {
        int i = 25;
        int j = i+++1;
        if (j % 5 == 0) {
            System.out.println(j + " is divisible by 5");
        } else {
            System.out.println(j + " is not divisible by 5");
        }
        System.out.println("Done");
    }
}
```

What is the result?

A. Compilation fails.

B. 25 is divisible by 5
   Done

C. 26 is not divisible by 5
   Done

D. 27 is not divisible by 5
   Done

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!

QUESTION 60
Given the code fragment:

```
class Test {
    public static void main(String[] args) {
        Test t = new Test();
        int[] arr = new int[10];
        arr = t.subArray(arr, 0, 2);
    }
    // insert code fragment here
}
```

Which method definition can be inserted at line 7 to enable the code to compile?

A. ```
public int[] subArray(int[] src, int start, int end) {
    return src;
    }
```
B. ```
public int subArray(int src, int start, int end) {
    return src;
    }
```
C. ```
public int[] subArray(int src, int start, int end) {
    return src;
    }
```
D. ```
public int subArray(int[] src, int start, int end) {
    return src;
    }
```

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
No explanation yet!