1z0-803.exam.125q

Number: 1z0-803
Passing Score: 800
Time Limit: 120 min

https://www.gratisexam.com/

1z0-803

Java SE 7 Programmer I

Exam A

QUESTION 1
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.

Incorrect answers:
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 compares values for equality.

There is no generic comparison method named same.

= = (with a space) is not a valid method.

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 2
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.

Incorrect answers:
int[2][2] array2D;
The size of the array cannot be defined this way.

int array2D[]; is good definition of a one-dimensional array.
int[] []array2D[];is good definition of a three-dimensional array.

QUESTION 3
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 4
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
```

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

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.

Incorrect answers:
B: The second println statement would not run.
C: The first println statement will always run.

QUESTION 5
Which two may precede the word 'class' in a class declaration?

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

Correct Answer: BC
Section: (none)
Explanation

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 6
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 ArrayIndexoutofBoundsException 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:

QUESTION 7
Given:

```
5.  // insert code here
6.      public abstract void bark();
7.  }
8.
9.  // insert code here
10.     public void bark() {
11.         System.out.println("woof");
12.     }
13. }
```

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. class 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 {
```

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

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 8

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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 9
Given:

```
public static void main(String[] args) {

     int a, b, c = 0;
     int a, b, c;
     int g, int h, int i = 0;
     int d, e, F;
     Int k, l, m = 0;
```

Which two declarations will compile?

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

Correct Answer: AD
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Incorrect answers:
int a, b, c;
OK, but duplicate definition.

QUESTION 10
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 11
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.

Incorrect answers:
StringBuilder sb = new String("128");
StringBuilder not String is required.

setCapacity or getInstance do not work.

QUESTION 12
Given:

```
public class DoCompare4 {
    public static void main(String[] args) {
        String[] table = {"aa", "bb", "cc"};
        int ii = 0;
        do
            while (ii < table.length)
                System.out.println(ii++);
        while (ii < table.length);
    }
}
```

What is the result?

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

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 13
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 14
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 15
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.println( "..." );
```

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 16
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 17
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 ma

Correct Answer: B
Section: (none)
Explanation

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

QUESTION 18
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:

QUESTION 19
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 real match
B. They really match
C. They match
D. Nothing is printed to the screen

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 20
Given the following code:

```
1.  public class Simple {
2.  public float price;
3.  public static void main(String[] args) {
4.  Simple price = new Simple();
5.  price = 4;
6.  }
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 21
Given:

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

What is the result?

A. 2
   1
B. 2
   1
   0
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 22
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 23
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 24
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 25
Give:

```
public class MyFive {
    public 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 26
Given the code fragment:

Boolean b1 = true;
Boolean b2 = false;
int i = 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.

Incorrect answers:
the compareTo method produces and int, not a boolean.
i = 1 is an assignment, not a comparison.
i == 2? -1:0 would produce the integer 0. A Boolean value is needed.

QUESTION 27
Given:

```
public class SuperTest {
    public static void main(String args[]) {
        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();
square.foo("bar");
```

C)
```
Square square = new Square();
square.foo();
square.foo("bar");
```

D)
```
Square square = new Square();
square.foo("bar");
square.foo();
```

E)
```
Square square = new Square();
square.foo();
square.foo();
```

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

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

Explanation:
Incorrect answers:
A: Produces
Shape: constructor
Square: constructor
Square: foo
Shape: foo

B: Produces
Shape: constructor
Square: constructor
Square: foo
Square: foo

C: Produces
Shape: constructor
Shape: foo
Square: foo

E: Produces
Shape: constructor
Shape: foo
Shape: foo

QUESTION 28
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. Java.util
D. Javax.net
E. Java.*
F. The package with no name

Correct Answer: AF
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.

Incorrect answers:
Java.awt: basic hierarchy of packages for native GUI components
Javax.net: networking operations, sockets, DNS lookups, etc.

QUESTION 29
Given:

```
1.  public abstract class  Wow {
2.      private int wow;
3.      public Wow{int wow) {
4.          this.wow = wow;
5.      }
6.      public void wow() { }
7       private void wowza() { }
8.  }
```

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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 30
Given:

```
class X {
    static void m(int i) {
        i += 7;
    }
    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:

QUESTION 31
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:
http://docs.oracle.com/javase/tutorial/java/IandI/abstract.html

QUESTION 32
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:

QUESTION 33
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 of java.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 34
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 35
View the Exhibit.

```
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

```
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:

QUESTION 36

```
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
D. 1
   2

Correct Answer: AB
Section: (none)
Explanation

Explanation/Reference:
Explanation:
A: Output is 2 if Math.random() is greater than 0.5.
B: If Math.random() returns a value less equal to 0.5, the code won't throw an exception, it will continue with the doMore() method which will println "4" after which the program will continue with the doStuff() method and will println "3", after that we will be back in main() and the program will print "1".

QUESTION 37
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:

QUESTION 38
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:

QUESTION 39
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:

QUESTION 40

```
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:

QUESTION 41
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:

QUESTION 42
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:

QUESTION 43
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:

QUESTION 44
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 45
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:

QUESTION 46

```
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 order to produce output 2345?

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

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 47

```
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: 33
B. i: 15
j: 33
C. i: 33
j: 23
D. i: 15
j: 23

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

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 48

```
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:

QUESTION 49
Which statement will empty 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:

QUESTION 50

```
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:

QUESTION 51
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:

QUESTION 52

```
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:

QUESTION 53
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:

QUESTION 54
Which three statements are benefits of encapsulation?

A. Allows a class implementation to change without changing t he 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:

QUESTION 55
The protected modifier on a Field declaration within a public class means that the field _____.

A. Cannot be modified
B. Can be read but not written from outside the class
C. Can be read and written from this class and its subclasses only within the same package
D. Can be read and written from this class and its subclasses defined in any package

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Reference:
http://beginnersbook.com/2013/05/java-access-modifiers/

QUESTION 56
Given:

```
class Caller {
    private void init() {
        System.out.println(”Initialized”);
    }

    public void start() {
        init();
        System.out.println(”Started”);
    }
}

public class TestCall {
    public static void main(String[] args) {
        Caller c = new Caller();
        c.start();
        c.init();
    }
}
```

What is the result?

A. Initialized
   Started
B. Initialized
   Started
   Initialized
C. Compilation fails
D. An exception is thrown at runtime

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 57

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 ";
    }
}
```

A. X XX
B. X Y X
C. Y Y X
D. Y YY

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 58
Given:

```
class Alpha {
    int ns;
    static int s;
    Alpha(int ns) {
        if (s < ns) {
            s = ns;
            this.ns = ns;
        }
    }
    void doPrint() {
        System.out.println("ns = " + ns + " s = " + s);
    }
```

And

```
public class TestA {
    public static void main (String[] args) {
        Alpha ref1 = new Alpha(50);
        Alpha ref2 = new Alpha(125);
        Alpha ref3 = new Alpha(100);
        ref1.doPrint();
        ref2.doPrint();
        ref.doPrint();
    }
}
```

A. ns = 50 S = 125
   ns = 125 S = 125
   ns = 100 S = 125
B. ns = 50 S = 125
   ns = 125 S = 125
   ns = 0 S = 125
C. ns = 50 S = 50
   ns = 125 S = 125
   ns = 100 S = 100
D. ns = 50 S = 50

ns = 125 S = 125
ns = 0 S = 125

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 59
Given:

```
Class A { }
Class B { }

Interface X { }
Interface Y { }
```

Which two definitions of class C are valid?

A. Class C extends A implements X { }
B. Class C implements Y extends B { }
C. Class C extends A, B { }
D. Class C implements X, Y extends B { }
E. Class C extends B implements X, Y { }

Correct Answer: AE
Section: (none)
Explanation

Explanation/Reference:
extends is for extending a class.

implements is for implementing an interface.
Java allows for a class to implement many interfaces.

QUESTION 60

Given the code fragment:

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

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

A. this.fvar = 200; this.cvar = 400;
B. t.fvar = 200; Test2.cvar = 400;
C. cvar = 400;
D. t.fvar = 200;
E. fvar = 200; cvar = 400;
F. this.fvar = 200; Test2.cvar = 400;

Correct Answer: BCD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 61
View the exhibit.

```
class MissingInfoException extends Exception { }

class AgeOutofRangeException extends Exception { }

class Candidate {
    String name;
    int age;
    Candidate(String name, int age) throws Exception {
        if (name == null) {
            throw new MissingInfoException();
        } else if (age <= 10 || age >= 150) {
            throw new AgeOutofRangeException();
        } else {
            this.name = name;
            this.age = age;
        }
    }
    public String toString() {
        return name + " age: " + age;
    }
}
```

Given the code fragment:

```
4.  public class Test {
5.      public static void main(String[] args) {
6.          Candidate c = new Candidate(”James”, 20);
7.          Candidate c1 = new Candidate(”Williams”, 32);
8.          System.out.println(c);
9.          System.out.println(c1);
10.     }
11. }
```

Which change enables the code to print the following?

James age: 20
Williams age: 32

A. Replacing line 5 with public static void main (String [] args) throws MissingInfoException, AgeOutofRangeException {
B. Replacing line 5 with public static void main (String [] args) throws.Exception {
C. Enclosing line 6 and line 7 within a try block and adding:
   catch(Exception e1) { //code goes here}
   catch (missingInfoException e2) { //code goes here}
   catch (AgeOutofRangeException e3) {//code goes here}
D. Enclosing line 6 and line 7 within a try block and adding:
   catch (missingInfoException e2) { //code goes here}
   catch (AgeOutofRangeException e3) {//code goes here}

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 62
Given:

```
public class Test {

     static void dispResult (int[] num) {
          try {
               System.out.println(num[1] / (num[1] - num[2]));
          } Catch(ArithmeticException e) {
               System.err.println("fisrt exception");
          }
          System.out.println("Done");
     }

     public static void main(String[] args) {
          try {
               int[] arr = {100, 100);
               dispResult(arr);
          } catch(IllegalArgumentException e) {
               System.err.println("second exception");
          } catch(Exception e) {
               System.err.println("third exception");
          }
     }
}
```

What is the result?

A. 0
   Done
B. First Exception
   Done
C. Second Exception
D. Done
   Third Exception
E. Third Exception

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 63
Given the code format:

```
class DBConfiguration {
     String user;
     String password;
}
```

And:

```
4.   public class DBHandler {
5.        DBConfiguration configureDB(String uname, String password) {
6.             // insert code here
7.        }
8.        public static void main(String[] args) {
9.             DBHandler r = new DBHandler();
10.            DBConfiguration dbConf = r.ConfigureDB("manager", "manager");
11.       }
12.  }
```

Which code fragment must be inserted at line 6 to enable the code to compile?

A. DBConfiguration f;
   return f;
B. Return DBConfiguration;
C. Return new DBConfiguration;
D. Return 0;

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 64
Given:

```
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.println("Red " + obj.sum);
          obj.doCheck(2);
          System.out.println("Orange " + obj.sum);
          obj.doCheck(3);
          System.out.println("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
D. Green 4
E. Compilation fails

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:

QUESTION 65
Given the code fragment:

```
String color = "teal";

switch (color) {
     case "Red":
          System.out.println("Found Red");
     case "Blue":
          System.out.println("Found Blue");
          break;
     case "Teal":
          System.out.println("Found Teal");
          break;
     default:
          System.out.println("Found Default");
}
```

What is the result?

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

E. Found Default

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 66
Given:

```
class X {
    public void mX() {
        System.out.println("Xml");
    }
}
class Y extends X {
    piblic 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();
    }
}
```

A. Ym
Xm2

B. Ym
Xm1
C. Compilation fails
D. A ClassCastException is thrown at runtime

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 67
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

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 68
Given:

```
public class MyFor1 {
    public static void main(String[] args) {
        int [] x = {6, 7, 8};
        for (int i : x) {
                System.out.print(i + " ");
        }
    }
}
```

What is the result?

A. 6 7 8
B. 7 8 9
C. 0 1 2
D. 6 8 10
E. Compilation fails

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 69
Given:

```
public class Calculator {
    public static void main(String[] args) {
        int num = 5;
        int sum;

        do {
            sum += num;
        } 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:

QUESTION 70
Given:

```
package p1;
public interface DoInterface {
    void m1(int n);
    public void m2(int n);
}

package p3;
import p1.DoInterface;
public class DoClass implements DoInterface{
    int x1,x2;
    DoClass(){
        this.x1 = 0;
        this.x2 = 10;
    }
    public void m1(int p1) { x1+=p1; System.out.println(x1); } // line n2
    public void m2(int p1) { x2+=p1; System.out.println(x2); }
}

package p2;
import p1.*;
import p3.*;
class Test {
    public static void main(String[] args){
        DoInterface doi = new DoClass();  // line n3
        doi.method1(100);
        doi.method2(200);
    }
}
```

What is the result?

A. 100
   2 10
B. Compilation fails due to an error in line n1
C. Compilation fails due to an error at line n2
D. Compilation fails due to an error at line n3

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 71
Given:

```
public class TestTry {
    public static void main(String[] args) {
        StringBuilder message = new StringBuilder("hello java!");
        int pos = 0
        try {
            for ( pos = 0 < 12; pos++) {
                switch (message.charAt (pos)) {
                    case 'a':
                    case 'e':
                    case 'o':
                        String uc=Character.toString(message.charAt(pos)).toUpperCase();
                        message.replace(pos, pos+1, uc);
                    }
            }
        } catch (Exception e) {
            System.out.println("out of limits");
        }
        System.out.println(message);
    }
}
```

What is the result?

A. hEllOjAvA!
B. Hello java!
C. Out of limits
   hEllOjAvA!
D. Out of limits

Correct Answer: C

Section: (none)
Explanation

Explanation/Reference:

QUESTION 72
Given:

```
public class App {
    public static void main(String[] args) {
        int i = 10;
        int j = 20;
        int k = j += i / 5;
        System.out.print(i + " : " + j + " : " + k);
    }
}
```

What is the result?

A. 10 : 22 : 20
B. 10 : 22 : 22
C. 10 : 22 : 6
D. 10 : 30 : 6

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 73
Given the code fragment:

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

What is the result?

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

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 74
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. The program prints nothing
B. d
C. A StringIndexOutOfBoundsException is thrown at runtime.
D. AnArrayIndexOutOfBoundsException is thrown at runtime.
E. A NullPointerException is thrown at runtime.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 75
Which two statements are true for a two-dimensional array of primitive data type?

A. It cannot contain elements of different types.
B. The length of each dimension must be the same.
C. At the 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: CD
Section: (none)
Explanation

Explanation/Reference:
http://stackoverflow.com/questions/12806739/is-an-array-a-primitive-type-or-an-object-or-something-else-entirely

QUESTION 76
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;
    }
}
```

A. Option A

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

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 77
Given:

```
public class MyFor3 {
    public static void main(String[] args) {
        int[] xx = null;
        for (int ii : xx) {
            System.out.println(ii);
        }
    }
}
```

What is the result?

A. Null
B. Compilation fails
C. An exception is thrown at runtime
D. 0

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 78
Given:

```
public class Test3 {
    public static void main(String[] args) {
        String names[] = new String[3];
        names[0] = "Mary Brown";
        names[1] = "Nancy Red";
        names[2] = "Jessy Orange";
        try {
            for(String n: names) {
                try {
                    String pwd = n.substring(0, 3)+n.substring(6,10);
                    System.out.println(pwd);
                }
                catch(StringIndexOutOfBoundsException sie) {
                    System.out.println("string out of limits");
                }
            }
        }
        catch(ArrayIndexOutOfBoundsException e) {
            System.out.println("array out of limits");
        }
    }
}
```

What is the result?

A. Marrown
String out of limits
JesOran

B. Marrown
String out of limits
Array out of limits

C. Marrown

String out of limits

D. Marrown
NanRed
JesOran

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 79
Given:

```
public abstract class Shape {
    private int x;
    private int y;
    public abstract void draw();
    public void setAnchor(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

Which two classes use the shape class correctly?

A)
```
public class Circle implements Shape {
    private int radius;
}
```
B)
```
public abstract class Circle extends Shape {
    private int radius;
}
```
C)
```
public class Circle extends Shape {
    private int radius;
    public void draw();
}
```
D)
```
public abstract class Circle implements Shape {
    private int radius;
    public void draw();
}
```
E)
```
public class Circle extends Shape {
    private int radius;
    public void draw(){/* code here */};
}
```
F)
```
public abstract class Circle implements Shape {
    private int radius;
    public void draw(){/* code here */};
}
```

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

Correct Answer: BE
Section: (none)
Explanation

Explanation/Reference:
When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class (E). However, if it does not, then the subclass must also be declared abstract (B).
Note: An abstract class is a class that is declared abstract—it may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclassed.

QUESTION 80
Given the class definitions:

```
class Alpha {
    public String doStuff(String msg) {
        return msg;
    }
}
class Beta extends Alpha {
    public String doStuff(String msg) {
        return msg.replace('a', 'e');
    }
}
class Gamma extends Beta {
    public String doStuff(String msg) {
        return msg.substring(2);
    }
}
```

And the code fragment of the main() method,

```
12. List<Alpha> strs = new ArrayList<Alpha>();
13. strs.add(new Alpha());
14. strs.add(new Beta());
15. strs.add(new Gamma());
16. for (Alpha t : strs) {
17.     System.out.println(t.doStuff("Java"));
18. }
```

What is the result?

A. Java
   Java
   Java
B. Java
   Jeve
   va
C. Java
   Jeve
   ve
D. Compilation fails

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 81
Given:

```
public class Msg {
    public static String doMsg(char x) {
        return "Good Day!";
    }
    public static String doMsg(int y) {
        return "Good Luck!";
    }
    public static void main(String[] args) {
        char x = 8;
        int z = '8';
        System.out.println(doMsg(x));
        System.out.print(doMsg(z));
    }
}
```

What is the result?

A. Good Day!
Good Luck!
B. Good Day!
Good Day!
C. Good Luck!
Good Day!
D. Good Luck!
Good Luck!
E. Compilation fails

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 82
Which two items can legally be contained within a java class declaration?

A. An import statement
B. A field declaration
C. A package declaration
D. A method declaration

Correct Answer: BD
Section: (none)
Explanation

Explanation/Reference:
Reference:
http://docs.oracle.com/javase/tutorial/java/javaOO/methods.html

QUESTION 83
Given:

```
public class Palindrome {
    public static int main(String[] args) {
        System.out.print(args[1]);
        return 0;
    }
}

And the commands:
javac Palindrome.java
java Palindrome Wow Mom
```

What is the result?

A. Compilation fails
B. The code compiles, but does not execute.
C. Paildrome
D. Wow
E. Mom

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 84
Given:

```
class Jump {
    static String args[] = {"lazy", "lion", "is", "always");
    public static void main(String[] args) {
        System.out.println(
            args[1] + " " + args[2] + " " + args[3] + " jumping");
    }
}
```

And the commands:

Javac Jump.java
Java Jump crazy elephant is always

What is the result?

A. Lazy lion is jumping
B. Lion is always jumping
C. Crazy elephant is jumping
D. Elephant is always jumping
E. Compilation fails

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 85
Which code fragment cause a compilation error?

A. flat flt = 100F;
B. float flt = (float) 1_11.00;
C. float flt = 100;
D. double y1 = 203.22;
   floatflt = y1
E. int y2 = 100;

floatflt = (float) y2;

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 86
Given:

```
class Test {
    public static void main(String[] args) {
        int numbers[];
        numbers = new int[2];
        numbers[0] = 10;
        numbers[1] = 20;

        numbers = new int[4];
        numbers[2] = 30;
        numbers[3] = 40;
        for (int x : numbers) {
            System.out.print(" "+x);
        }
    }
}
```

What is the result?

A. 10 20 30 40

B. 0 0 30 40
C. Compilation fails
D. An exception is thrown at runtime

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 87
Given:

```
public class Series {
    public static void main(String[] args) {
        int arr[] = {1, 2, 3,};

        for (int var : arr) {
            int i = 1;
            while (i <= var);
                System.out.println(i++);
        }
    }
}
```

What is the result?

A. 1
   1
   1
B. 1
   2
   3

C. 2
   3
   4
D. Compilation fails
E. The loop executes infinite times

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:

QUESTION 88
Give:

```
class Alpha {
    public String[] main = new String[2];
    Alpha(String[] main) {
        for (int ii = 0; ii < main.length; ii++) {
            this.main[ii] = main[ii] +5;
        }
    }
    public void main() {
    System.out.print(main[0] + main[1]);
    }
}

public class Test {
    public static void main(String[] args) {
        Alpha main = new Alpha(args);
        main.main();
    }
}
```

And the commands:

```
javac Test.java
java Test 1 2
```

What is the result?

A. 1525
B. 13

C. Compilation fails
D. An exception is thrown at runtime
E. The program fails to execute due to runtime error

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 89
Given:

```
public class Test {
    public static void main(String[] args) {
        Test ts = new Test();
        System.out.print(isAvailable + " ");
        isAvailable = ts.doStuff();
        System.out.println(isAvailable);
    }
    public static boolean doStuff() {
        return !isAvailable;
    }
    static boolean isAvailable = false;
}
```

What is the result?

A. true true
B. true false
C. false true

D. false false
E. Compilation fails

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:

QUESTION 90
Given the code in a file Traveler.java:

```
class Tours {
    public static void main(String[] args) {
        System.out("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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 91
Given:

```
class X {
    int x1, x2, x3;
}
class Y extends X {
    int y1;
    Y() {
        x1 = 1;
        x2 = 2;
        y1 = 10;
    }
}

class 2 extends Y {
    int z1;
    z() {
        x1 = 3;
        y1 = 20;
        z1 = 100;
    }
}
```

And,

```
public class Test3 {
    public static void main(String[] args) {
        Z obj = new Z();
        System.out.println(obj.x3 + ", " + obj.y1 + ", " + obj.z1);
    }
}
```

Which constructor initializes the variable x3?

A. Only the default constructor of class X
B. Only the no-argument constructor of class Y

C. Only the no-argument constructor of class Z
D. Only the default constructor of object class

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 92
Given:

```
public class CharToStr {
    public static void main(String[] args) {
        String str1 = "Java";
        char str2[] = { 'J', 'a', 'v', 'a' };
        String str3 = null;
            for (char c : str2) {
                str3 = str3 + c;
            }
            if (str1.equals(str3))
                System.out.print("Successful");
            else
                System.out.print("Unsuccessful");
    }
}
```

What is result?

A. Successful
B. Unsuccessful

C. Compilation fails
D. An exception is thrown at runtime

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 93
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:

QUESTION 94
Given the fragment:

```
24.  float var1 = (12_345.01 >= 123_45.00) ? 12_456 : 124_56.02f);
25.  float var2 = var1 + 1024;
26.  System.out.print(var2);
```

What is the result?

A. –13480.0
B. 13480.02
C. Compilation fails
D. An exception is thrown at runtime

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 95
Given the code fragment:

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

What is the result?

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

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 96
Given the classes:

* AssertionError
* ArithmeticException
* ArrayIndexOutofBoundsException
* FileNotFoundException
* IllegalArgumentException
* IOError

* IOException
* NumberFormatException
* SQLException

Which option lists only those classes that belong to the unchecked exception category?

A. AssertionError, ArrayIndexOutOfBoundsException, ArithmeticException
B. AssertionError, IOError, IOException
C. ArithmeticException, FileNotFoundException, NumberFormatException
D. FileNotFoundException, IOException, SQLException
E. ArrayIndexOutOfBoundException, IllegalArgumentException, FileNotFoundException

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Not B: IOError and IOException are both checked errors.
Not C, not D, not E: FileNotFoundException is a checked error.
Note:
Checked exceptions:
* represent invalid conditions in areas outside the immediate control of the program (invalid user input, database problems, network outages, absent files)
* are subclasses of Exception
* a method is obliged to establish a policy for all checked exceptions thrown by its implementation (either pass the checked exception further up the stack, or handle it somehow)

Note:
Unchecked exceptions:
* represent defects in the program (bugs) - often invalid arguments passed to a non-private method. To quote from The Java Programming Language, by Gosling, Arnold, and Holmes: "Unchecked runtime exceptions represent conditions that, generally speaking, reflect errors in your program's logic and cannot be reasonably recovered from at run time."
* are subclasses of RuntimeException, and are usually implemented using IllegalArgumentException, NullPointerException, or IllegalStateException
* method is not obliged to establish a policy for the unchecked exceptions thrown by its implementation (and they almost always do not do so)

QUESTION 97
Given:

```
public class Test1 {
    static void doubling (Integer ref, int pv) {
        ref =20;
        pv = 20;
    }
    public static void main(String[] args) {
        Integer iObj = new Integer(10);
        int iVar = 10;
        doubling(iObj++, iVar++);
        System.out.println(iObj+ ", "+iVar);
```

What is the result?

A. 11, 11
B. 10, 10
C. 21, 11
D. 20, 20
E. 11, 12

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
The code doubling(iObj++, iVar++); increases both variables from to 10 to 11.

QUESTION 98
Given:

```
import java.util.*;
public class Ref {
    public static void main(String[] args) {
        StringBuilder s1 = new StringBuilder("Hello Java!");
        String s2 = s1.toString();
        List<String> lst = new ArrayList<String>();
        lst.add(s2);
        System.out.println(s1.getClass());
        System.out.println(s2.getClass());
        System.out.println(lst.getClass());
    }
}
```

What is the result?

A. class java.lang.String
   class java.lang.String
   class java.util.ArrayList
B. class java.lang.Object
   class java.lang. Object
   class java.util.Collection
C. class java.lang.StringBuilder
   class java.lang.String
   class java.util.ArrayList
D. class java.lang.StringBuilder
   class java.lang.String
   class java.util.List

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
class java.lang.StringBuilder
class java.lang.String
class java.util.ArrayList

QUESTION 99
Given:

```
public class ComputeSum {
    public int x;
    public int y;
    public int sum;
    public ComputeSum (int nx, int ny) {
        x = nx; y =ny;
        updateSum();
           }
    public void setX(int nx) { x = nx; updateSum();}
    public void setY(int ny) { x = ny; updateSum();}
  void updateSum() { sum = x + y;}
}
```

This class needs to protect an invariant on the sum field.

Which three members must have the private access modifier to ensure that this invariant is maintained?

A. The x field
B. The y field
C. The sum field
D. The ComputerSum ( ) constructor
E. The setX ( ) method
F. The setY ( ) method

Correct Answer: CEF
Section: (none)
Explanation

Explanation/Reference:
The sum field and the two methods (setX and SetY) that updates the sum field.

QUESTION 100
Given the following four Java file definitions:

```
// Foo.java
package facades;
public interface Foo { }

// Boo.java
package facades;
public interface Boo extends Foo { }

// Woofy.java
package org.domain
// line n1
public class Woofy implements Boo, Foo { }

// Test.java
package.org;
public class Test {
public static void main(String[] args) {
Foo obj=new Woofy();
```

Which set modifications enable the code to compile and run?

A. At line n1, Insert: import facades;
   At line n2, insert:
   import facades;
   import org.domain;
B. At line n1, Insert: import facades.*;
   At line n2, insert:
   import facades;
   import org.*;
C. At line n1, Insert: import facades.*;
   At line n2, insert:
   import facades.Boo;
   import org.*;
D. At line n1, Insert: import facades.Foo, Boo;
   At line n2, insert:
   import org.domain.Woofy;
E. At line n1, Insert: import facades.*;

At line n2, insert:
import facades;
import org.domain.Woofy;

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:

QUESTION 101
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 obj1 = null;
          obj2.num = 60;
          graceMarks(obj2);
     }
}
```

How many objects are created in the memory runtime?

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

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
obj1 and obj3.

when you do e2 = e1 you're copying object references - you're not making a copy of the object - and so the variables e1 and e2 will both point to the same object.

QUESTION 102
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:
1200 Strawberry
1230 Chocolate

QUESTION 103
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:

QUESTION 104
Given:

```
class Base {
    // insert code here
}
public class Derived extends Base{
    public static void main(String[] args) {
 Derived obj = new Derived();
 obj.setNum(3);
 System.out.println("Square = " + obj.getNum() * obj.getNum());
    }
}
```

Which two options, when inserted independently inside class Base, ensure that the class is being properly encapsulated and allow the program to execute and print the square of the number?

A.
```
private int num;
    public int getNum() {
        return num;
    }
public void setNum(int num) {
    this.num = num;
}
```

B.
```
public int num;

    protected public int getNum() {
        return num;
    }
protected public void setNum(int num) {
    this.num = num;
}
```

C.
```
private int num;

public int getNum() {
return num;
}
 private void setNum(int num) {
    this.num = num;
}
```

D.
```
protected int num;

     public int getNum() {
        return num;
    }
 public void setNum(int num) {
    this.num = num;
}
```

E.
```
protected int num;

     private int getNum() {
        return num;
    }
 public void setNum(int num) {
    this.num = num;
}
```

Correct Answer: AD

Section: (none)
Explanation

Explanation/Reference:
Incorrect:
Not B: illegal combination of modifiers: protected and public
not C: setNum method cannot be private.
not E: getNum method cannot be private.

QUESTION 105
Given:

```
public class Test {
    public static void main(String[] args) {
        int arr[] = new int[4];
        arr[0] = 1;
        arr[1] = 2;
        arr[2] = 4;
        arr[3] = 5;
        int sum = 0;
        try {
            for (int pos = 0; pos <= 4; pos++) {
                sum = sum + arr[pos];
             }
        } catch (Exception e) {
            System.out.println("Invalid index");
        }
        System.out.println(sum);
    }
}
```

What is the result?

A. 12
B. Invalid Index
12
C. Invalid Index
D. Compilation fails

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
The loop (for (int pos = 0; pos <= 4; pos++) {), it should be pos <= 3, causes an exception, which is caught. Then the correct sum is printed.

QUESTION 106
Given:

```
public class Equal {
    public static void main(String[] args) {
        String str1 = "Java";
        String[] str2 = {"J","a","v","a"};
        String str3 = "";
        for (String str : str2) {
            str3 = str3+str;
        }
        boolean b1 = (str1 == str3);
        boolean b2 = (str1.equals(str3));
        System.out.print(b1+", "+b2);
    }
```

What is the result?

A. true, false
B. false, true

C. true, true
D. false, false

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
== strict equality.
equals compare state, not identity.

QUESTION 107
Given the code fragment:

```
public static void main(String[] args) {
        int iArray[] = {65, 68, 69};
        iArray[2] = iArray[0];
        iArray[0] = iArray[1];
        iArray[1] = iArray[2];
        for (int element : iArray) {
            System.out.print(element + " ");
         }
}
```

A. 68, 65, 69
B. 68, 65, 65
C. 65, 68, 65
D. 65, 68, 69
E. Compilation fails

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
68 65 65

QUESTION 108
Given:

```
public class MyClass {
    public static void main(String[] args) {
        while (int ii = 0; ii < 2) {
        ii++;
        System.out.println("ii = " + ii);
    }
  }
}
```

What is the result?

A. ii = 1
   ii = 2
B. Compilation fails
C. The program prints nothing
D. The program goes into an infinite loop with no output
E. The program goes to an infinite loop outputting:
   ii = 1
   ii = 1

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
The while statement is incorrect. It has the syntax of a for statement.

The while statement continually executes a block of statements while a particular condition is true. Its syntax can be expressed as:

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

The while statement evaluates expression, which must return a boolean value. If the expression evaluates to true, the while statement executes the statement(s) in the while block. The while statement continues testing the expression and executing its block until the expression evaluates to false.

Reference: The while and do-while Statements

QUESTION 109
Given:

```
public class String1 {
    public static void main(String[] args) {
        String s = "123";
        if (s.length() >2)
            s.concat("456");
        for(int x = 0; x <3; x++)
            s += "x";
        System.out.println(s);
    }
}
```

What is the result?

A. 123
B. 123xxx
C. 123456
D. 123456xxx
E. Compilation fails

Correct Answer: B
Section: (none)

Explanation

Explanation/Reference:
123xxx

The if clause is not applied.

Note: Syntax of if-statement:
if (Statement) {
}

QUESTION 110
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: ABC
Section: (none)
Explanation

Explanation/Reference:
A: Private constructors prevent a class from being explicitly instantiated by its callers.
If the programmer does not provide a constructor for a class, then the system will always provide a default, public no-argument constructor. To disable this default constructor, simply add a private no-argument constructor to the class. This private constructor may be empty.
B: The following works fine:

```
int cake() {
    int cake=0;
    return (1);
 }
```

C: We can overload static method in Java. In terms of method overloading static method are just like normal methods and in order to overload static method you need to provide another static method with same name but different method signature.

Incorrect:
Not D: Only a public class in an application need to have a main method.

Not E:
Example:

```
class A
{
  public string something;
  public int a;
}
```

Q: What do you call classes without methods?
Most of the time: An anti pattern.

Why? Because it faciliates procedural programming with "Operator" classes and data structures. You separate data and behaviour which isn't exactly good OOP.

Often times: A DTO (Data Transfer Object)

Read only datastructures meant to exchange data, derived from a business/domain object.

Sometimes: Just data structure.

Well sometimes, you just gotta have those structures to hold data that is just plain and simple and has no operations on it.

Not F: Fields need to be initialized. If not the code will not compile.
Example:
Uncompilable source code - variable x might not have been initialized

QUESTION 111
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 grading process 89 50 104

What is the result?

A. Pass
B. Fail
C. Class MarketOutOfBoundsException
D. Class IndexOutOfBoundsException
E. Class Exception

Correct Answer: C

Section: (none)
Explanation

Explanation/Reference:
The value 104 will cause a MarketOutOfBoundsException

QUESTION 112
Given the code fragment:

```
StringBuilder sb = new StringBuilder ( ) ;
Sb.append ("world");
```

Which code 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);D

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
The java.lang.StringBuilder.insert(int offset, char c) method inserts the string representation of the char argument into this sequence.
The second argument is inserted into the contents of this sequence at the position indicated by offset. The length of this sequence increases by one. The offset argument must be greater than or equal to 0, and less than or equal to the length of this sequence.

Reference: Java.lang.StringBuilder.insert() Method

QUESTION 113
Given:

```
package p1;
public interface DoInterface {
    void method1(int n1);   // line n1
}
package p3;
import p1.DoInterface;
public class DoClass implements DoInterface {
    public DoClass(int p1) { }
    public void method1(int p1) {   } // line n2
    private void method2(int p1) {   } // line n3
}
public class Test {
    public static void main(String[] args) {
        DoInterface doi= new DoClass(100); // line n4
        doi.method1(100);
        doi.method2(100);
    }
}
```

Which change will enable the code to compile?

A. Adding the public modifier to the declaration of method1 at line n1
B. Removing the public modifier from the definition of method1 at line n2
C. Changing the private modifier on the declaration of method 2 public at line n3
D. Changing the line n4 DoClass doi = new DoClass ( );

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Private members (both fields and methods) are only accessible inside the class they are declared or inside inner classes. private keyword is one of four access modifier provided by Java and it's a most restrictive among all four e.g. public, default(package), protected and private.

Read more: http://javarevisited.blogspot.com/2012/03/private-in-java-why-should-you-always.html#ixzz3Sh3mOc4D

QUESTION 114
Given the fragment:

```
String[][] arra = new String[3][];
arra[0] = new String[]{"rose", "lily"};
arra[1] = new String[]{"apple", "berry","cherry","grapes"};
arra[0] = new String[]{"beans", "carrot","potato"};
// insert code fragment here
```

Which code fragment when inserted at line '// insert code fragment here', enables the code to successfully change array elements to uppercase?

A.
```
String[][] arra = new String[3][];
    arra[0] = new String[]{"rose", "lily"};
    arra[1] = new String[]{"apple", "berry","cherry","grapes"};
    arra[0] = new String[]{"beans", "carrot","potato"};
    for (int i = 0; i < arra.length; i++) {
        for (int j=0; j < arra[i].length; j++) {
            arra[i][j] = arra[i][j].toUpperCase();
        }
    }
```

B.
```
for (int i = 0; i < 3; i++) {
        for (int j=0; j < 4; j++) {
            arra[i][j] = arra[i][j].toUpperCase();
        }
    }
```

C.
```
for (String a[]:arra[][]) {
        for (String x:a[]) {
            x.toUpperCase();
        }
    }
```

D.
```
for (int i:arra.length) {
        for (String x:arra) {
            arra[i].toUpperCase();
        }
    }
```

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Incorrect:
not A: arra.length is 3, but the subarrays have 2, 3 and 4 elements. Index will be out of bound.
not B: The subarrys are of different lengths. Index will be out of bound.
not D: Compile error.

QUESTION 115
Given the code fragment:

```
public class Test {
    static String[][] arr =new String[3][];
    private static void doPrint() {
        //insert code here
    }
    public static void main(String[] args) {
      String[] class1 = {"A","B","C"};
      String[] class2 = {"L","M","N","O"};
      String[] class3 = {"I","J"};
      arr[0] = class1;
      arr[1] = class2;
      arr[2] = class3;
      Test.doPrint();
    }
}
```

Which code fragment, when inserted at line //insert code here, enables the code to print COJ?

A.
```
int i = 0;
    for (String[] sub: arr) {
       int j = sub.length -1;
       for (String str: sub) {
          System.out.println(str[j]);
          i++;
       }
    }
```

B.
```
private static void doPrint() {
   for (int i = 0;i < arr.length;i++) {
       int j = arr[i].length-1;
       System.out.print(arr[i][j]);
       }
```

```
        }
C.  int i = 0;
            for (String[] sub: arr[][]) {
                int j = sub.length;
                System.out.print(arr[i][j]);
                i++;
            }
D.  for (int i = 0;i < arr.length-1;i++) {
                int j = arr[i].length-1;
                System.out.print(arr[i][j]);
                i++;
            }
```

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Incorrect:
not A: The following line causes a compile error:
System.out.println(str[j]);
Not C: Compile erro line:
for (String[] sub: arr[][])
not D: Output: C

QUESTION 116
Given:

```
public class FieldInit {
    char c;
    boolean b;
    float f;
    void printAll() {
        System.out.println("c = " + c);
        System.out.println("c = " + b);
        System.out.println("c = " + f);
    }

    public static void main(String[] args) {
     FieldInit f = new FieldInit();
     f.printAll();
    }
}
```

What is the result?

A. c = null
   b = false
   f = 0.0F

B. c = 0
   b = false
   f = 0.0f

C. c = null
   b = true
   f = 0.0

D. c =
   b = false
   f = 0.0

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 117
Given the code fragment:

```
String[] cartoons = {"tom","jerry","micky","tom"};
int counter =0;

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:
Counter++ will be executed only once because of the else if constructs.

QUESTION 118
Given:

```
public 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: AC
Section: (none)
Explanation

Explanation/Reference:
The following will work fine:

```
public class Test {
    public static void main(String[] args) {
        int day = 1;
        switch (day) {
            case 7: System.out.print("Uranus"); break;
            case 6: System.out.print("Saturn"); break;
            case 1: System.out.print("Mercury"); break;
            case 2: System.out.print("Venus"); break;
            case 3: System.out.print("Earth"); break;
            case 4: System.out.print("Mars"); break;
            case 5: System.out.print("Jupiter"); break;
        }
}
}
```

QUESTION 119
Given:

```
public class Test {
    public static void main(String[] args) {
        try {
            String[] arr =new String[4];
            arr[1] = "Unix";
            arr[2] = "Linux";
            arr[3] = "Solarios";
            for (String var : arr) {
                System.out.print(var + " ");
            }
        } catch(Exception e) {
            System.out.print (e.getClass());
        }
    }
}
```

What is the result?

A. Unix Linux Solaris
B. Null Unix Linux Solaris
C. Class java.lang.Exception
D. Class java.lang.NullPointerException

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
null Unix Linux Solarios

The first element, arr[0], has not been defined.

QUESTION 120

Given the code fragment

```
int var1 = -5;
int var2 = var1--;
int var3 = 0;
if (var2 < 0) {
    var3 = var2++;
} else {
    var3 = --var2;
}
System.out.println(var3);
```

What is the result?

A. – 6
B. – 4
C. – 5
D. 5
E. 4
F. Compilation fails

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 121
Given:

```
public class X {
    static int i;
    int j;
    public static void main(String[] args) {
        X x1 = new X();
        X x2 = new X();
        x1.i = 3;
        x1.j = 4;
        x2.i = 5;
        x2.j = 6;
        System.out.println(
                x1.i + " "+
                x1.j + " "+
                x2.i + " "+
                x2.j);
    }
}
```

What is the result?

A. 3 4 5 6
B. 3 4 3 6
C. 5 4 5 6
D. 3 6 4 6

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 122
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:
In both Java and C#, a "default constructor" refers to a nullary constructor that is automatically generated by the compiler if no constructors have been defined for the class. The default constructor is also empty, meaning that it does nothing. A programmer-defined constructor that takes no parameters is also called a default constructor.

QUESTION 123
Given the code fragment?

```
public 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 here
}
```

Which method can be inserted at line // insert code here 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:

QUESTION 124
Given:

```
package p1;
public class Test {
   static double dvalue;
   static Test ref;
    public static void main(String[] args) {
        System.out.println(ref);
        System.out.println(dvalue);
     }
}
```

What is the result?

A. p1.Test.class
   0.0
B. <the summary address referenced by ref>
   0.000000

C. Null
0.0

D. Compilation fails

E. A NullPointerException is thrown at runtime

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
null
0.0

QUESTION 125
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:
1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5