TROYTEC TECHNOLOGIES

Troy Tec ™

Java SE 7 Programmer I

Exam: 1Z0-803

Edition: 4.0

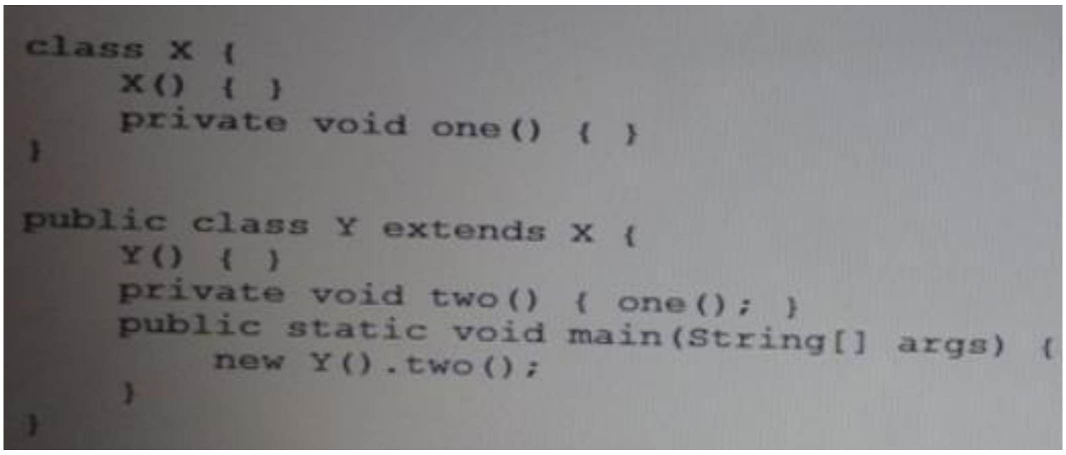
© 2014 - 2015 Troy Tec, LTD All Rights Reserved

QUESTION: 1
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

Answer: B

QUESTION: 2
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();
    }
}
```

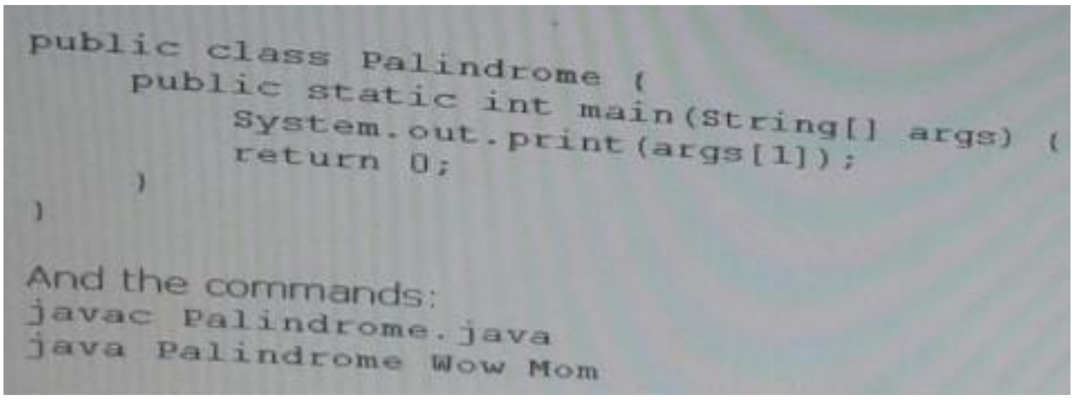
What changes will make this code compile? (Select Two)

A. Adding the public modifier to the declaration of class x
B. Adding the protected modifier to the x() constructor
C. Changing the private modifier on the declaration of the one() method to protected
D. Removing the Y () constructor
E. Removing the private modifier from the two () method

Answer: C, E

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: 3
Given:



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

And the commands:

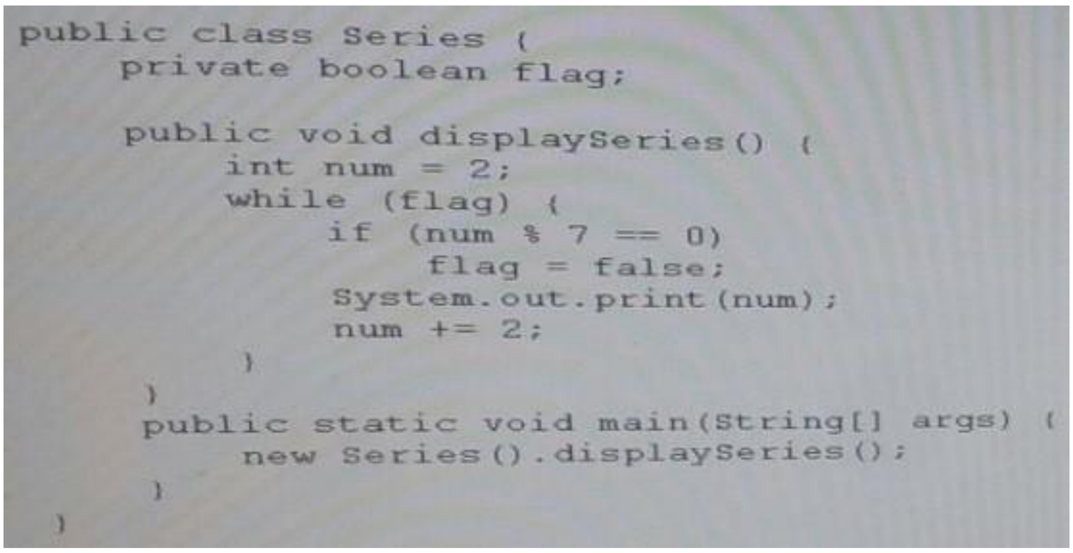
```
javac Palindrome.java
java Palindrome Wow Mom
```

What is the result?

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

Answer: B

QUESTION: 4
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();
    }
}
```

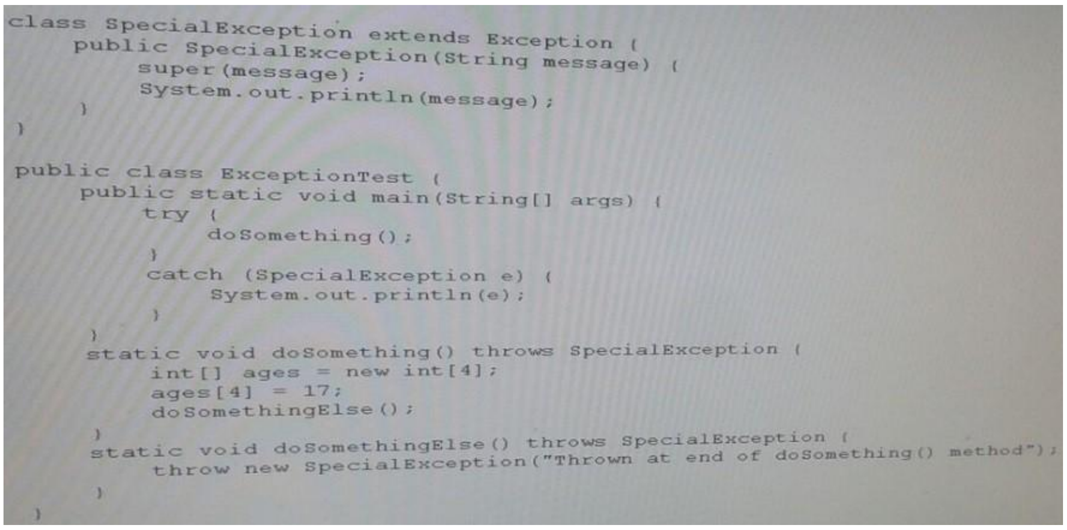
What is the result?

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

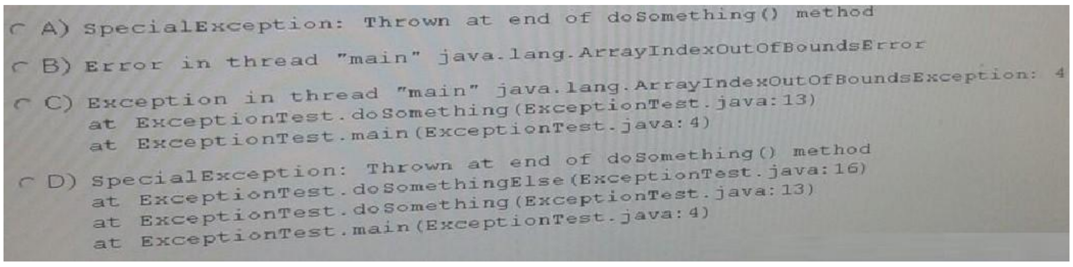
Answer: B

QUESTION: 5
Given:



```
class SpecialException extends Exception {
    public SpecialException(String message) {
        super(message);
        System.out.println(message);
    }
}

public class ExceptionTest {
    public static void main(String[] args) {
        try {
            doSomething();
        }
        catch (SpecialException e) {
            System.out.println(e);
        }
    }
    static void doSomething() throws SpecialException {
        int[] ages = new int[4];
        ages[4] = 17;
        doSomethingElse();
    }
    static void doSomethingElse() throws SpecialException {
        throw new SpecialException("Thrown at end of doSomething() method");
    }
}
```

What will be the output?



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

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

Answer: D

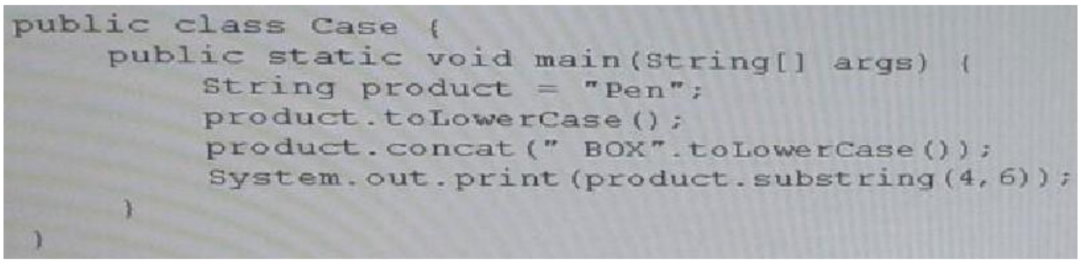
QUESTION: 6
Given:

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

Answer: D

QUESTION: 7
Given:



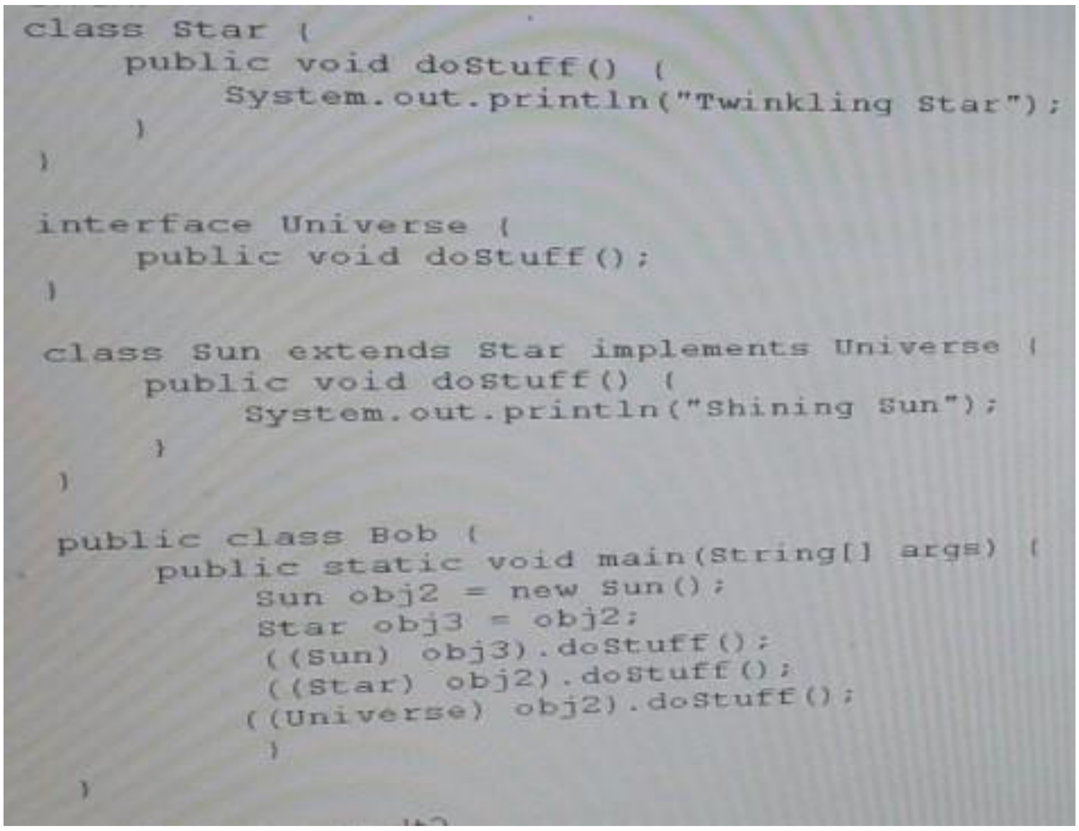
```
public class Case {
    public static void main(String[] args) {
        String product = "Pen";
        product.toLowerCase();
        product.concat(" BOX".toLowerCase());
        System.out.print(product.substring(4, 6));
    }
}
```

What is the result?

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

Answer: E

QUESTION: 8
Given:



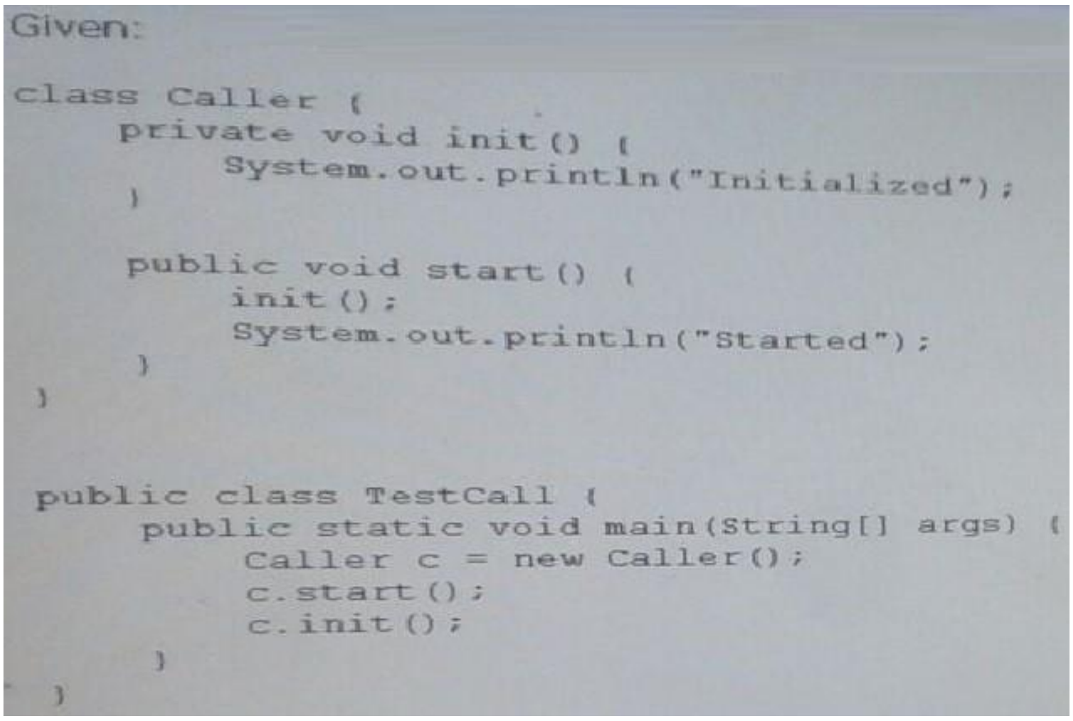
```
class Star {
    public void doStuff() {
        System.out.println("Twinkling Star");
    }
}

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

Answer: D

QUESTION: 9
Given:



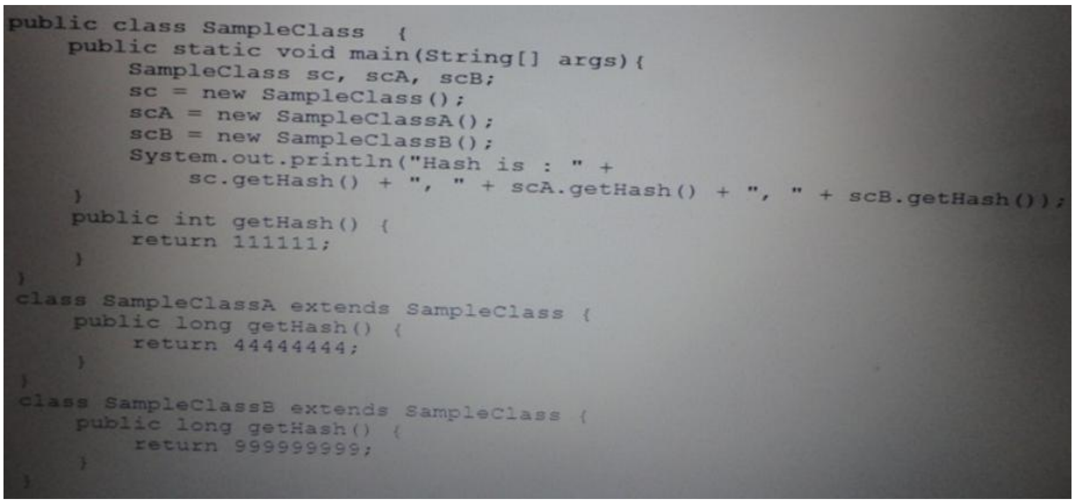
```
Given:

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

Answer: B

QUESTION: 10
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
D. Hash is: 111111, 44444444, 999999999

Answer: A

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: 11
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; }

Answer: D

QUESTION: 12
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);
}

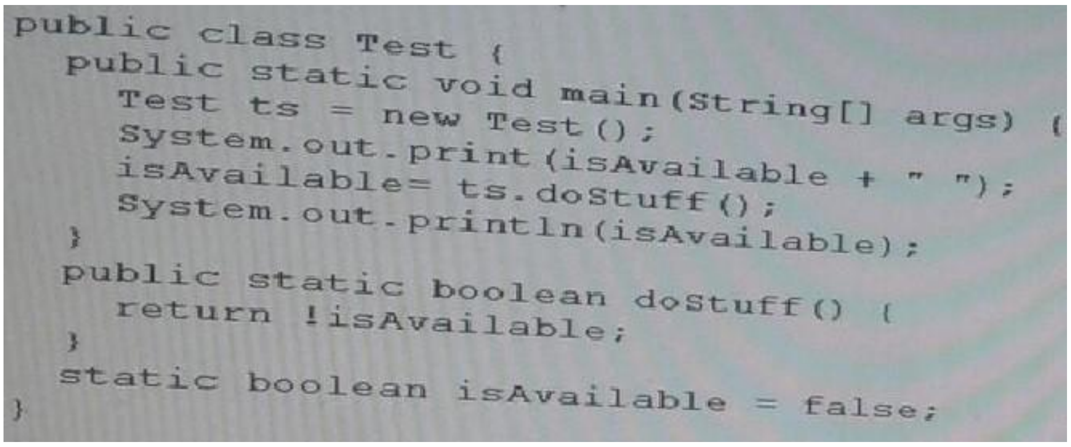
}
Which change will enable the code to compile?

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 ( );

Answer: C

Explanation:
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 its 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: 13
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

Answer: E

QUESTION: 14
Given the code fragment:
System.out.printIn("Result: " + 2 + 3 + 5);
System.out.printIn("Result: " + 2 + 3 * 5); What is the result?

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

Answer: C

Explanation:
First line:
System.out.println("Result: " + 2 + 3 + 5); String concatenation is produced.
Second line:
System.out.println("Result: " + 2 + 3 * 5);
3*5 is calculated to 15 and is appended to string 2. Result 215.
The output is: Result: 235
Result: 215
Note #1:
To produce an arithmetic result, the following code would have to be used:
System.out.println("Result: " + (2 + 3 + 5));
System.out.println("Result: " + (2 + 1 * 5)); run:
Result: 10
Result: 7
Note #2:
If the code was as follows:
System.out.println("Result: " + 2 + 3 + 5");
System.out.println("Result: " + 2 + 1 * 5");
The compilation would fail. There is an unclosed string literal, 5", on each line.

QUESTION: 15
Which two are valid array declaration?

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

Answer: A, C

QUESTION: 16
Given: 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 ("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();

Answer: F

QUESTION: 17
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

Answer: B

Explanation:
123xxx
The if clause is not applied.
Note: Syntax of if-statement: if (Statement) {}

QUESTION: 18
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

Answer: C

Explanation:
The value 104 will cause a MarketOutOfBoundsException

QUESTION: 19
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.

Answer: C

QUESTION: 20
Given:

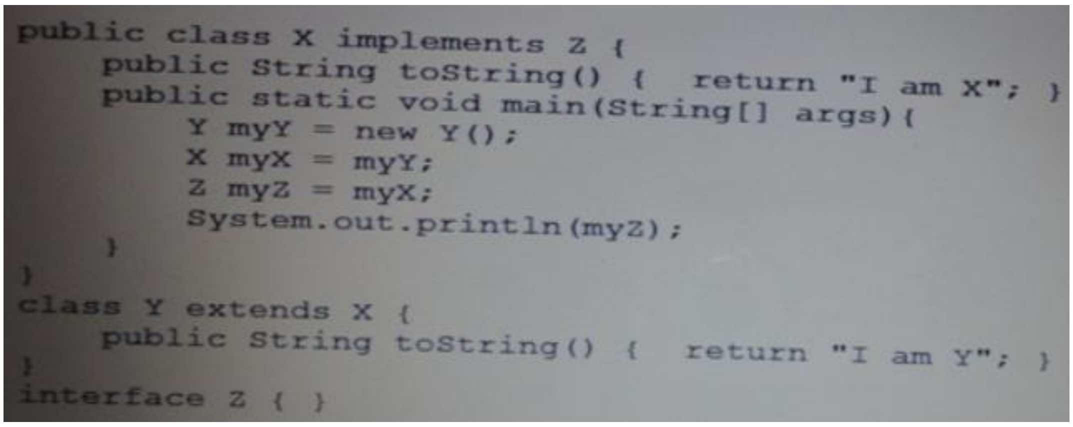
```
public class MyFor {
public static void main(String[] args) { for (int ii = 0; ii < 4; ii++) { System.out.println("ii =
"+ ii);
ii = ii +1;
}
}
}
```

What is the result?

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

Answer: A

QUESTION: 21
Given:



```
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.

Answer: C

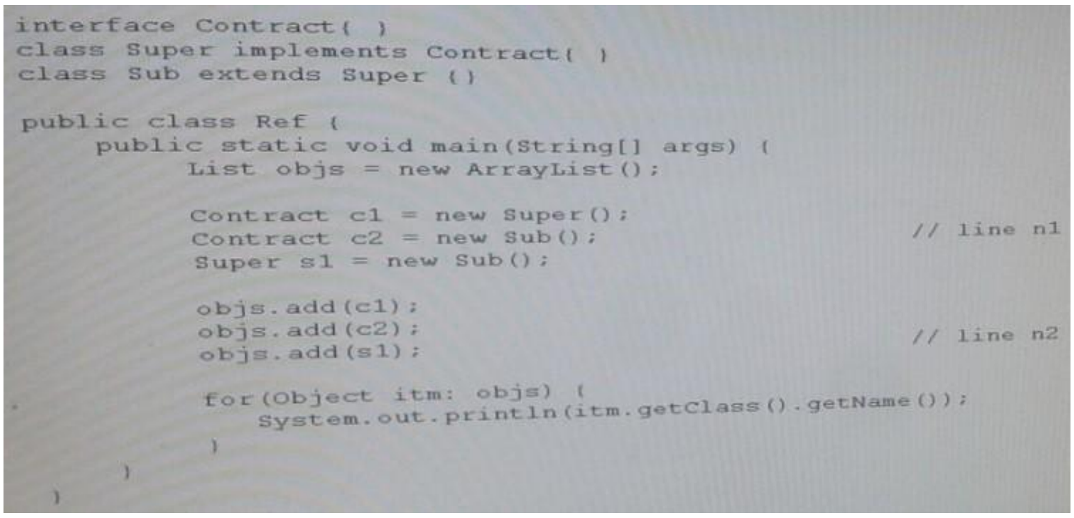
QUESTION: 22
Given:
public class TestOperator {
public static void main(String[] args) { int result = 30 -12 / (2*5)+1;
System.out.print("Result = " + result);
}
}
What is the result?

A. Result = 2
B. Result = 3
C. Result = 28
D. Result = 29
E. Result = 30

Answer: E

QUESTION: 23

Given the code fragments:



```
interface Contract{ }
class Super implements Contract{ }
class Sub extends Super {}

public class Ref {
     public static void main(String[] args) {
          List objs = new ArrayList();

          Contract c1 = new Super();
          Contract c2 = new Sub();                          // line n1
          Super s1 = new Sub();

          objs.add(c1);
          objs.add(c2);                                     // line n2
          objs.add(s1);

          for(Object itm: objs) {
              System.out.println(itm.getClass().getName());
          }
     }
}
```

What is the result?

A. Super Sub Sub
B. Contract Contract Super
C. Compilation fails at line n1
D. Compilation fails at line n2

Answer: D

QUESTION: 24

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

Answer: B

Explanation:
null Unix Linux Solarios
The first element, arr[0], has not been defined.

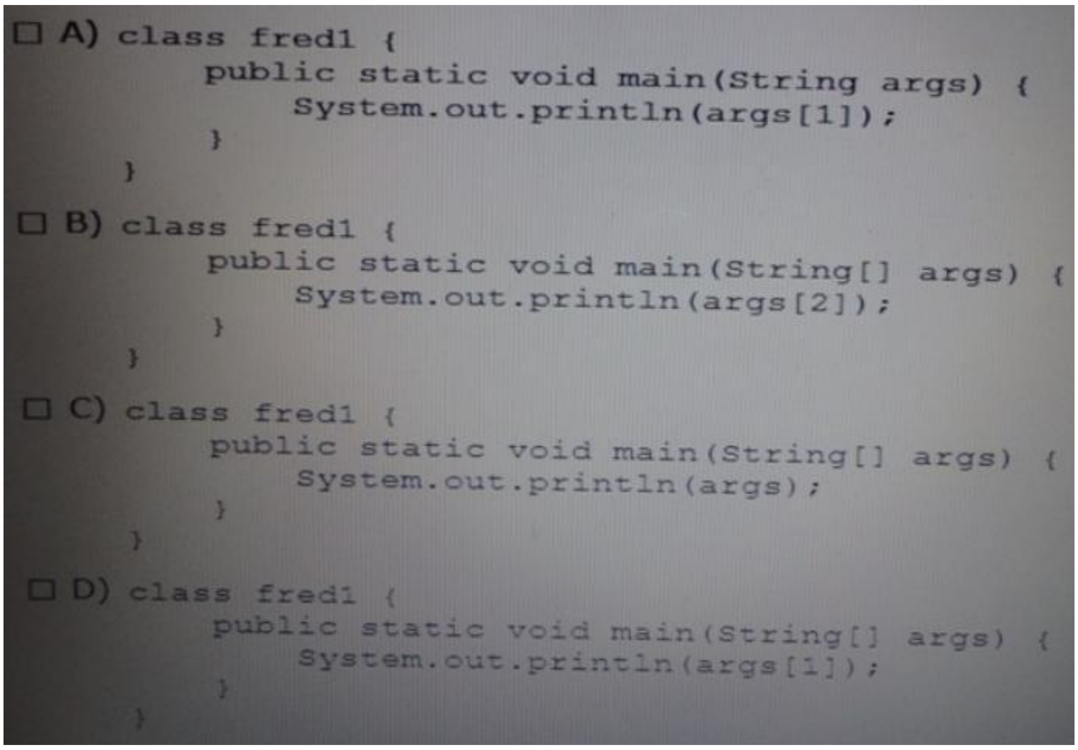
QUESTION: 25
Which two statements are true?

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.

Answer: A, E

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

QUESTION: 26
Which two will compile, and can be run successfully using the command: Java fred1 hello walls



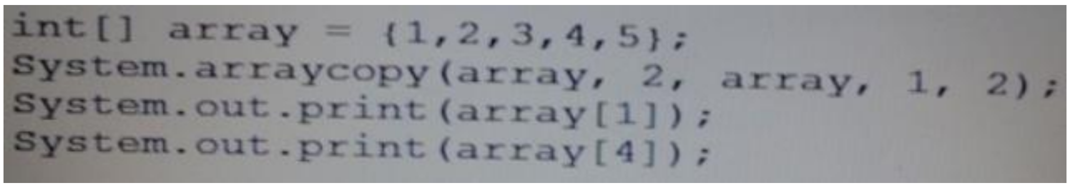
```
☐ A) class fred1 {
        public static void main(String args) {
            System.out.println(args[1]);
        }
    }
☐ B) class fred1 {
        public static void main(String[] args) {
            System.out.println(args[2]);
        }
    }
☐ C) class fred1 {
        public static void main(String[] args) {
            System.out.println(args);
        }
    }
☐ D) class fred1 {
        public static void main(String[] args) {
            System.out.println(args[1]);
        }
    }
```

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

Answer: C, D

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: 27
Given the fragment:



```
int[] array = {1,2,3,4,5};
System.arraycopy(array, 2, array, 1, 2);
System.out.print(array[1]);
System.out.print(array[4]);
```

What is the result?

A. 14
B. 15
C. 24
D. 25
E. 34
F. 35

Answer: F

Explanation:
The two elements 3 and 4 (starting from position with index 2) are copied into position index 1 and 2 in the same array. After the arraycopy command the array looks like:
{1, 3, 4, 4, 5};
Then element with index 1 is printed: 3 Then element with index 4 is printed: 5
Note: The System class has an arraycopy method that you can use to efficiently copy data from one array into another:
public static void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)
The two Object arguments specify the array to copy from and the array to copy to. The three int arguments specify the starting position in the source array, the starting position in the destination array, and the number of array elements to copy.

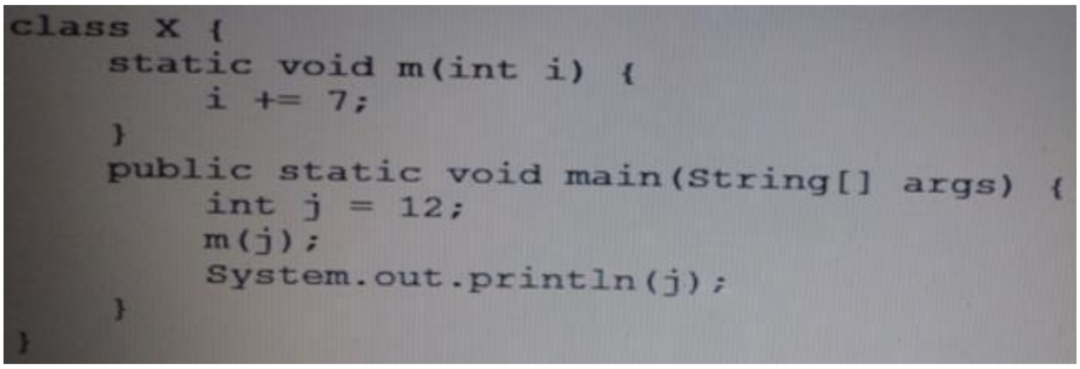
QUESTION: 28
A method doSomething () that has no exception handling code is modified to trail a method that throws a checked exception. Which two modifications, made independently, will allow the program to compile?

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

Answer: A, B

Explanation:
Valid Java programming language code must honor the Catch or Specify Requirement. This means that code that might throw certain exceptions must be enclosed by either of the following:
* A try statement that catches the exception. The try must provide a handler for the exception, as described in Catching and Handling Exceptions.
* A method that specifies that it can throw the exception. The method must provide a throws clause that lists the exception, as described in Specifying the Exceptions Thrown by a Method.
Code that fails to honor the Catch or Specify Requirement will not compile.

QUESTION: 29
Given:



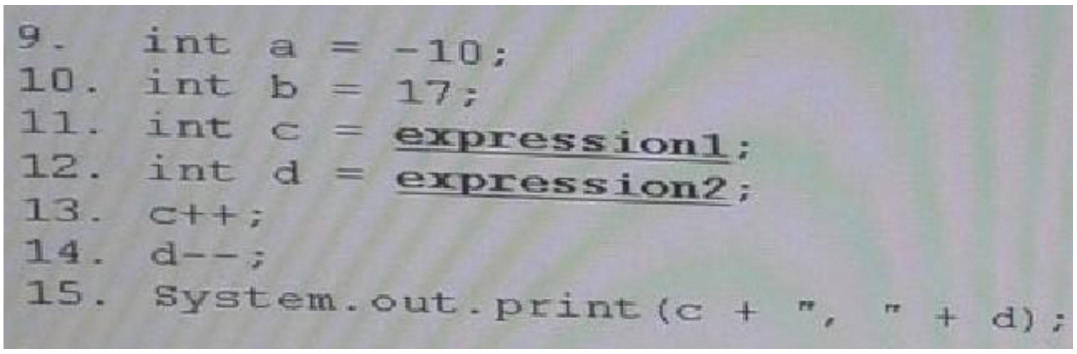
```
class X {
    static void m(int i) {
        i += 7;
    }
    public static void main(String[] args) {
        int j = 12;
        m(j);
        System.out.println(j);
    }
}
```

What is the result?

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

Answer: B

QUESTION: 30
Given the code fragment:



```
int a = -10;
int b = 17;
int c = expression1;
int d = expression2;
c++;
d--;
System.out.print(c + ", " + d);
```

What could expression1 and expression2 be, respectively, in order to produce output –8, 16?

A. + +a, - -b
B. + +a, b- -
C. A+ +, - - b
D. A + +, b - -

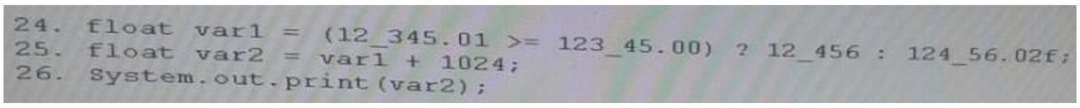
Answer: D

QUESTION: 31
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.

Answer: D

QUESTION: 32
Given the fragment:



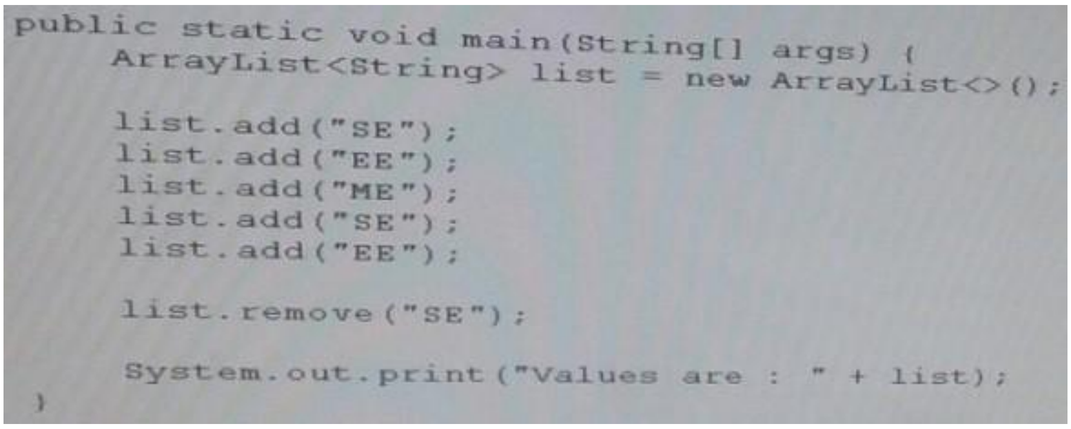
```
float var1 = (12_345.01 >= 123_45.00) ? 12_456 : 124_56.02f;
float var2 = var1 + 1024;
System.out.print(var2);
```

What is the result?

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

Answer: A

QUESTION: 33
Given the code fragment:



```
public static void main(String[] args) {
     ArrayList<String> list = new ArrayList<>();

     list.add("SE");
     list.add("EE");
     list.add("ME");
     list.add("SE");
     list.add("EE");

     list.remove("SE");

     System.out.print("Values are : " + list);
 }
```

What is the result?

A. Values are : [EE, ME]
B. Values are : [EE, EE, ME]
C. Values are : [EE, ME, EE]
D. Values are : [SE, EE, ME, EE]
E. Values are : [EE, ME, SE, EE]

Answer: E

QUESTION: 34
Which three are advantages of the Java exception mechanism?

A. Improves the program structure because the error handling code is separated from the normal program function
B. Provides a set of standard exceptions that covers all the possible errors
C. Improves the program structure because the programmer can choose where to handle exceptions
D. Improves the program structure because exceptions must be handled in the method in which they occurred
E. allows the creation of new exceptions that are tailored to the particular program being

Answer: A, C, E

Explanation:
A: The error handling is separated from the normal program logic. C: You have some choice where to handle the exceptions.
E: You can create your own exceptions.

QUESTION: 35
A method is declared to take three arguments. A program calls this method and passes only

two arguments. What is the result?

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

Answer: A

Explanation:
The problem is noticed at build/compile time. At build you would receive an error message like: required: int,int,int found: int,int

QUESTION: 36
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;

Answer: A, D

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: 37
Given the classes:
* AssertionError
* ArithmeticException
* ArrayIndexOutofBoundsException
* FileNotFoundException
* IllegalArgumentException
* IOError
* IOException
* NumberFormatException
* SQLException

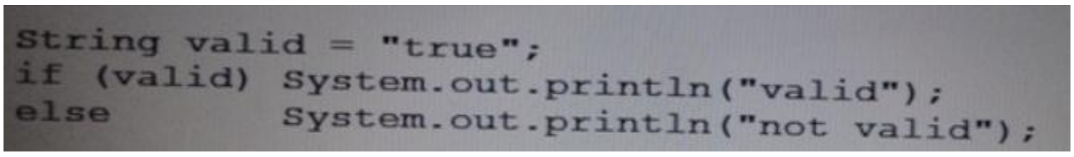
Which option lists only those classes that belong to the unchecked exception category?

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

Answer: A

Explanation:
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: 38
Given the code fragment:



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

What is the result?

A. Valid
B. Not valid
C. Compilation fails

D. An IllegalArgumentException is thrown at run time

Answer: C

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

QUESTION: 39
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;

Answer: B

QUESTION: 40
Given the code fragment:
// insert code here arr[0] = new int[3]; arr[0][0] = 1;
arr[0][1] = 2;
arr[0][2] = 3;
arr[1] = new int[4]; arr[1][0] = 10;
arr[1][1] = 20;
arr[1][2] = 30;
arr[1][3] = 40;
Which two statements, when inserted independently at line // insert code here, 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];

Answer: C, E

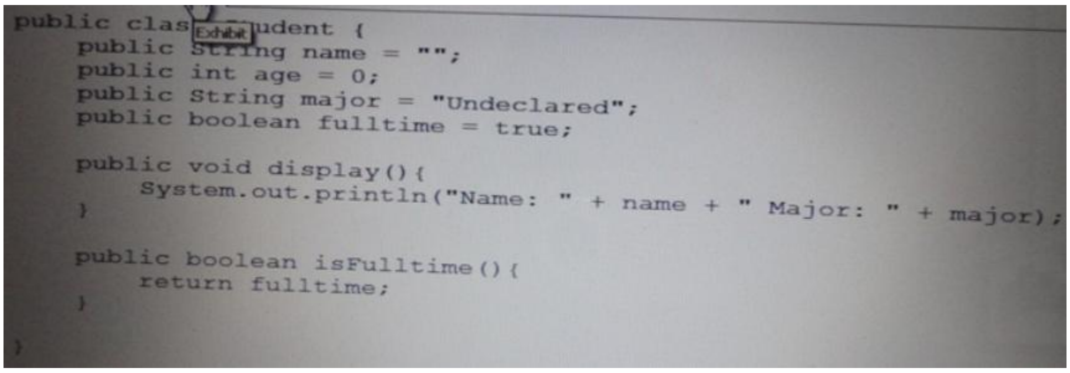
QUESTION: 41

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

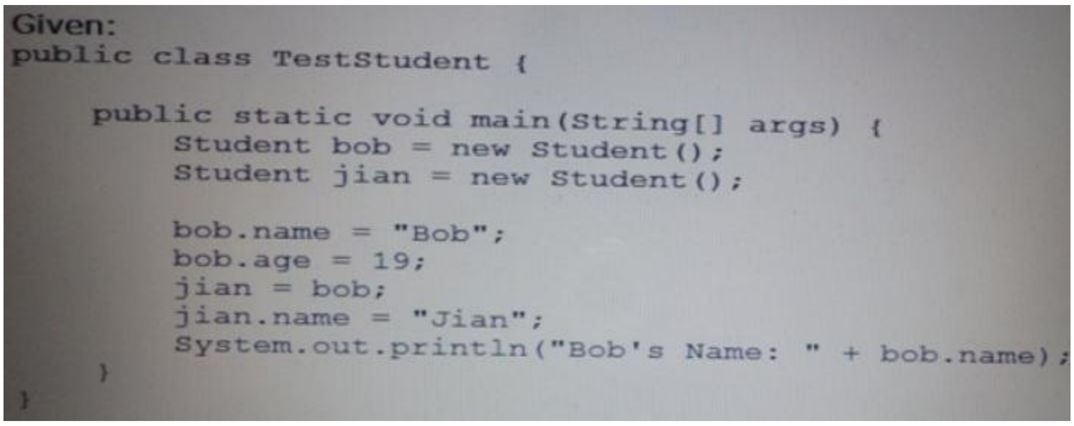
A. Load statement: array2d(x, y) = x + y;
Modify statement: array2d(x, y) = array2d(x, y) * 2
B. Load statement: array2d[x y] = x + y;
Modify statement: array2d[x y] = array2d[x y] * 2
C. Load statement: array2d[x, y] = x + y;
Modify statement: array2d[x, y] = array2d[x, y] * 2
D. Load statement: array2d[x][y]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

Answer: D

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();
        Student jian = new Student();

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

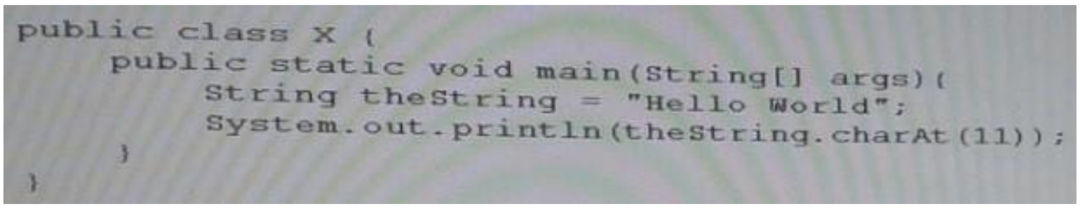
What is the result when this program is executed?

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

Answer: B

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

QUESTION: 43
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.

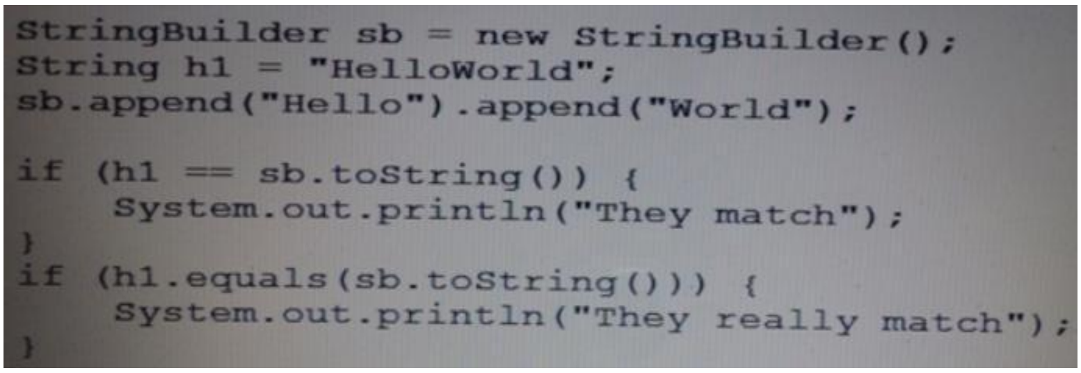
Answer: C

QUESTION: 44
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 mina(Object[] args) { System.out.println("four");
}
}
What is printed out when the program is excuted?

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

Answer: C

QUESTION: 45
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

Answer: B

QUESTION: 46
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 { }

Answer: A, E

Explanation:
extends is for extending a class.
implements is for implementing an interface.
Java allows for a class to implement many interfaces.

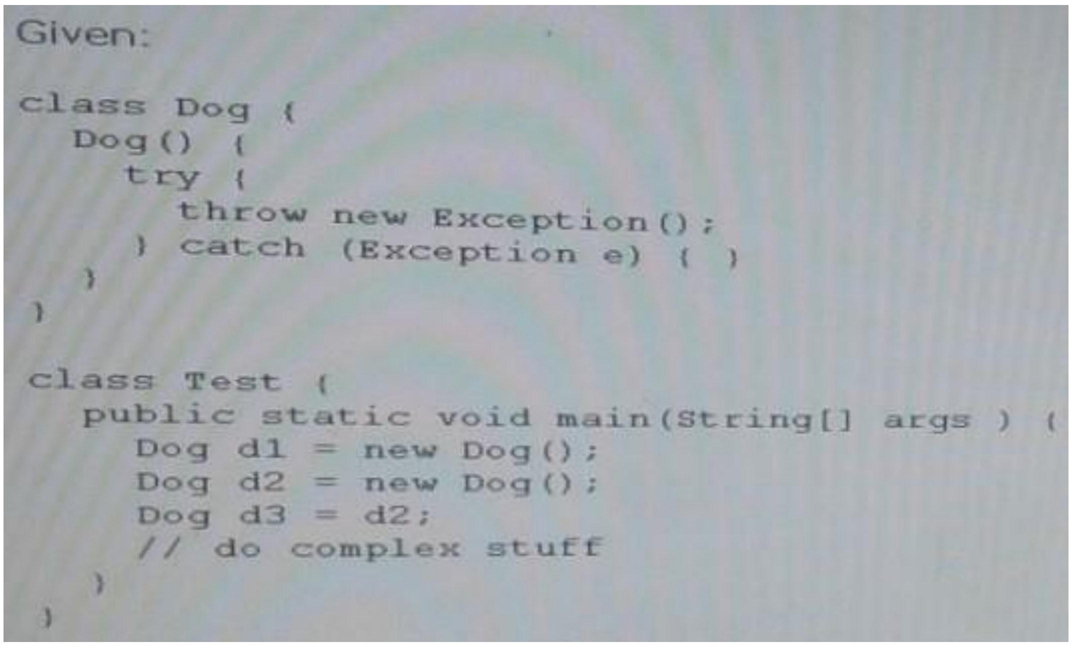
QUESTION: 47
Given:
public class Test1 {
static void doubling (Integer ref, int pv) { ref =20;
pv = 20;
}
public static void main(String[] args) { Integer iObj = new Integer(10);
int iVar = 10; doubling(iObj++, iVar++);
System.out.println(iObj+ ", "+iVar); What is the result?

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

Answer: A

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

QUESTION: 48
Given:


Given:

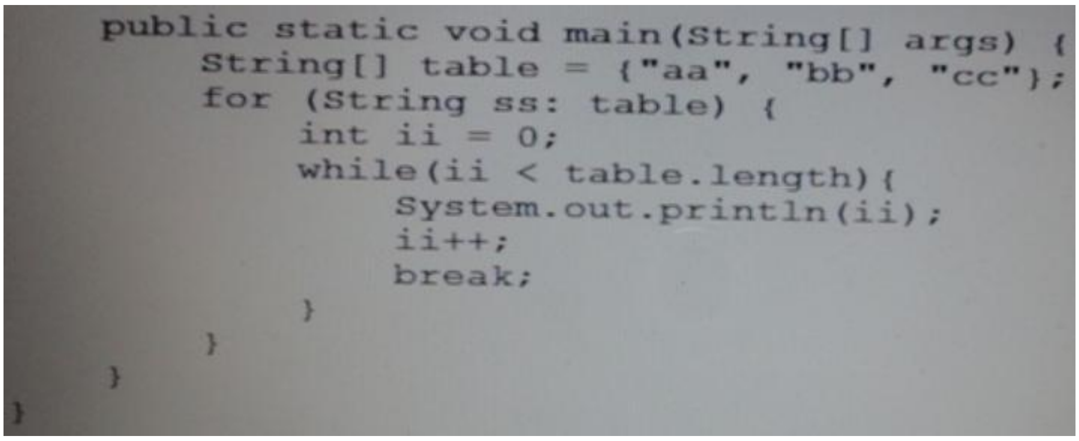
```
class Dog {
   Dog() {
      try {
         throw new Exception();
      } catch (Exception e) { }
   }
}

class Test {
   public static void main(String[] args ) {
      Dog d1 = new Dog();
      Dog d2 = new Dog();
      Dog d3 = d2;
      // do complex stuff
   }
}
```

How many objects have been created when the line / / do complex stuff is reached?

A. Two
B. Three
C. Four
D. Six

Answer: C

QUESTION: 49
Given the code fragment:



```
      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(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

Answer: B

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: 50
Given:

```
public class MyClass {
public static void main(String[] args) { String s = " Java Duke ";
int len = s.trim().length(); System.out.print(len);
}
}
```

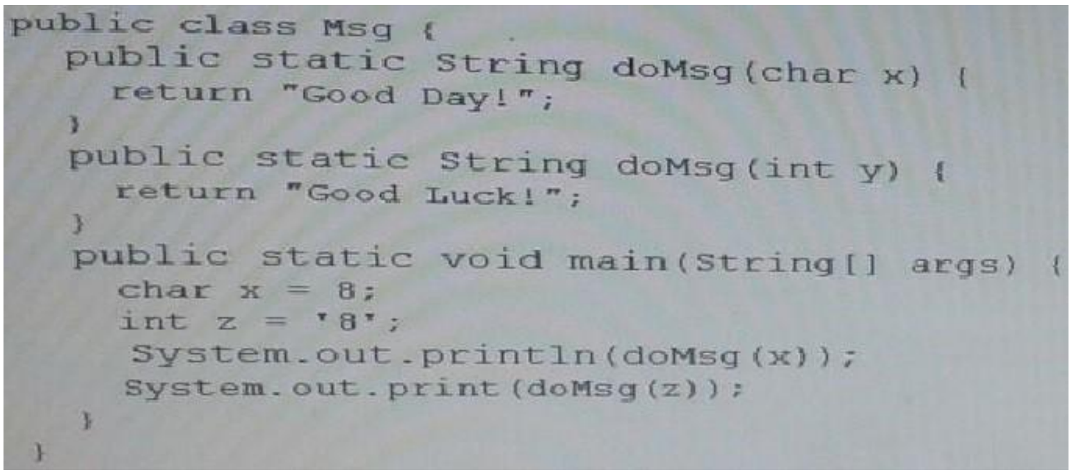
What is the result?

A. 8
B. 9
C. 11
D. 10
E. Compilation fails

Answer: B

Explanation:
Java - String trim() Method
This method returns a copy of the string, with leading and trailing whitespace omitted.

QUESTION: 51
Given:



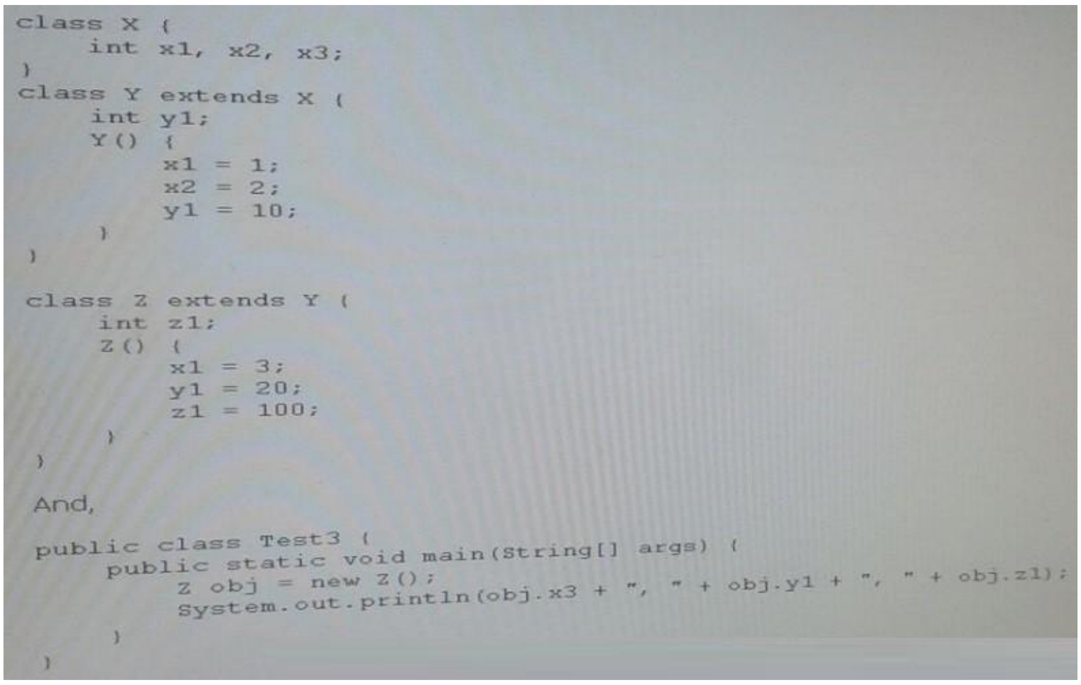
```
public class Msg {
    public static String doMsg(char x) {
        return "Good Day!";
    }
    public static String doMsg(int y) {
        return "Good Luck!";
    }
    public static void main(String[] args) {
        char x = 8;
        int z = '8';
        System.out.println(doMsg(x));
   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

Answer: E

QUESTION: 52
Given:



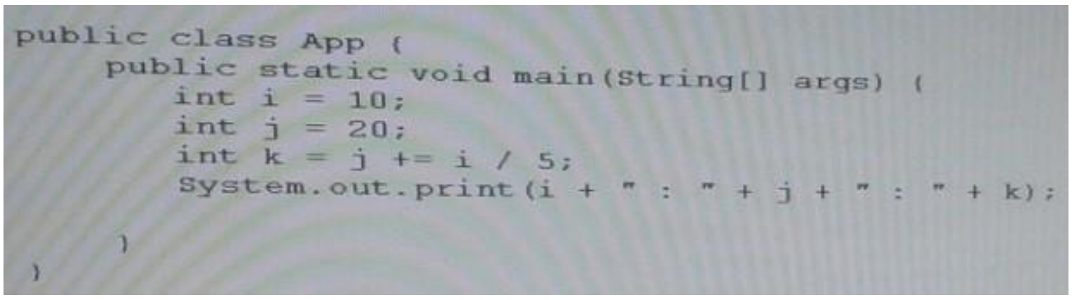
```
class X {
    int x1, x2, x3;
}
class Y extends X {
    int y1;
    Y() {
        x1 = 1;
        x2 = 2;
        y1 = 10;
    }
}

class Z 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

Answer: C

QUESTION: 53
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

Answer: B

QUESTION: 54
Which two may precede the word ‘class’ in a class declaration?

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

Answer: B, C

Explanation:
B: A class can be declared as public or private.
C: You can declare two kinds of classes: top-level classes and inner classes.
You define an inner class within a top-level class. Depending on how it is defined, an inner class can be one of the following four types: Anonymous, Local, Member and Nested top-level.
A nested top-level class is a member classes with 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: 55
Given the code fragment:

```
class Student { int rollnumber; String name;
List cources = new ArrayList();
// insert code here public String toString() {
return rollnumber + " : " + name + " : " + cources;
}
}
```

And,

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

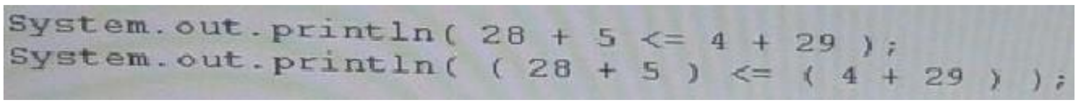
Which code fragment, when inserted at line // insert code here, enables class Test to print 123 : Fred : [Java, C]?

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 */
}

Answer: C

Explanation: I
ncorrect:
Not A: Student has private access line: Student s = new Student(123,"Fred", cs);
Not D: Cannot be applied to given types. Line: Student s = new Student(123,"Fred", cs);

QUESTION: 56
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

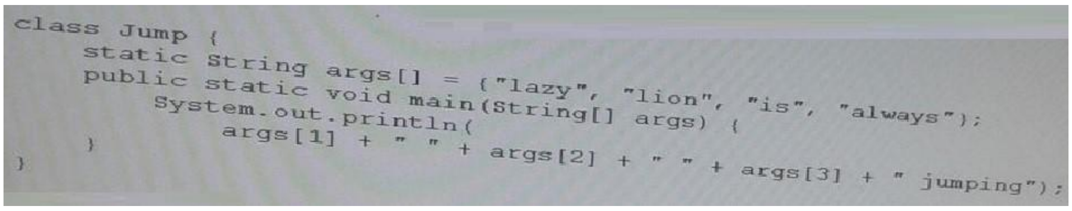
Answer: C

QUESTION: 57
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!

Answer: B

QUESTION: 58
Given:



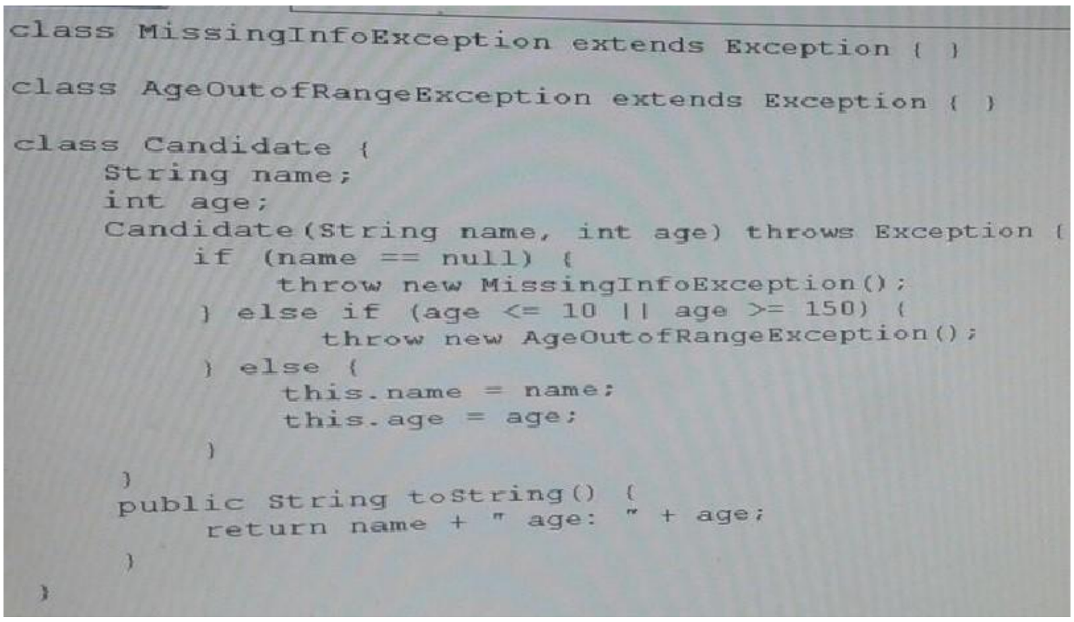
```
class Jump {
    static String args[] = {"lazy", "lion", "is", "always"};
    public static void main(String[] args) {
        System.out.println(
            args[1] + " " + args[2] + " " + args[3] + " jumping");
    }
}
```

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

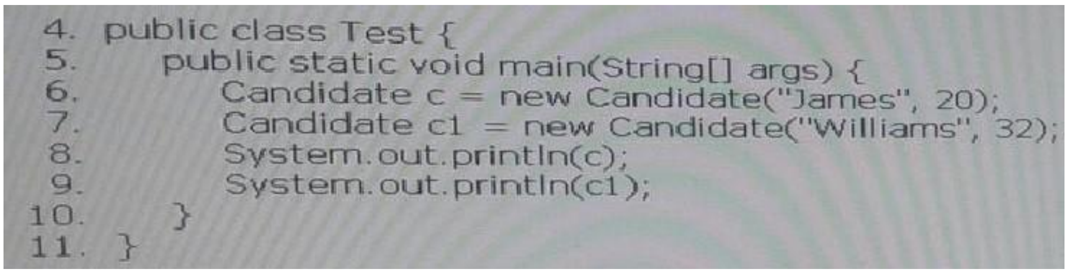
Answer: B

QUESTION: 59
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:



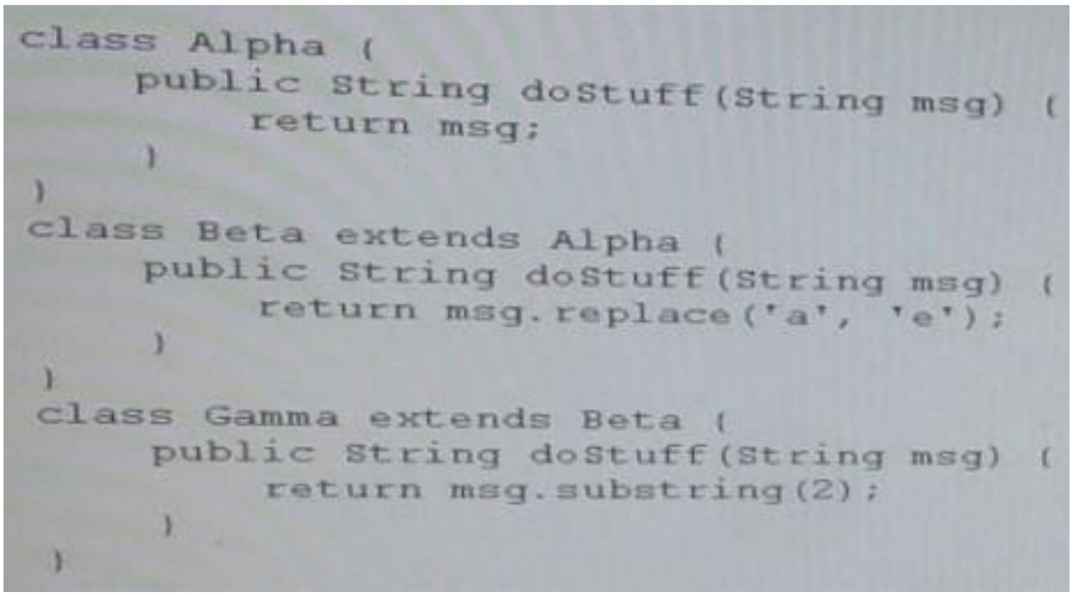
```
4.  public class Test {
5.      public static void main(String[] args) {
6.          Candidate c = new Candidate("James", 20);
7.          Candidate c1 = new Candidate("Williams", 32);
8.          System.out.println(c);
9.          System.out.println(c1);
10.     }
11. }
```

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}

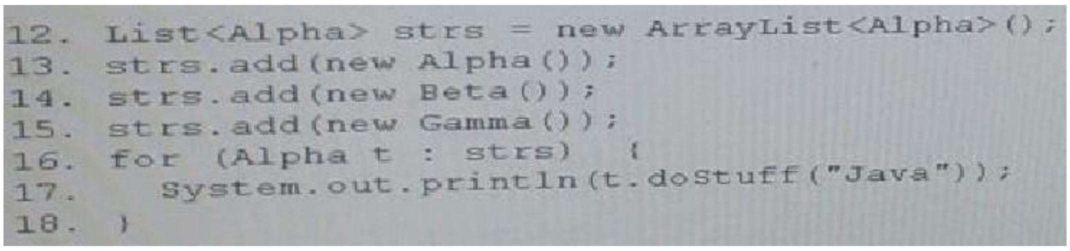
Answer: C

QUESTION: 60
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,



```
List<Alpha> strs = new ArrayList<Alpha>();
strs.add(new Alpha());
strs.add(new Beta());
strs.add(new Gamma());
for (Alpha t : strs)  {
    System.out.println(t.doStuff("Java"));
}
```

What is the result?

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

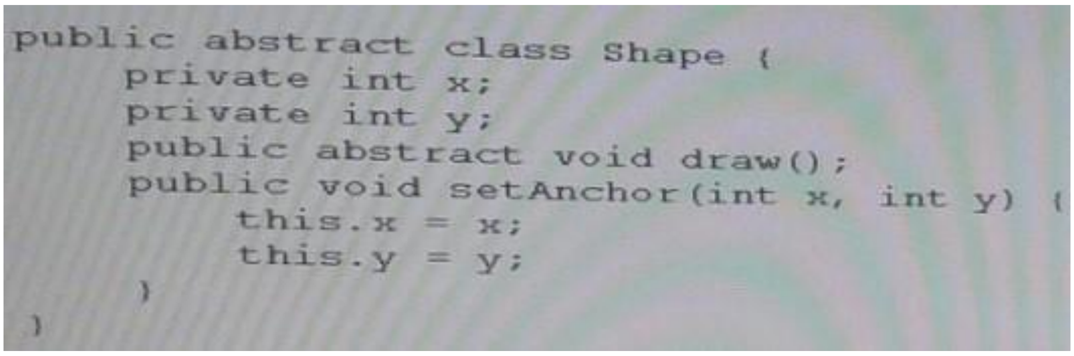
Answer: D

QUESTION: 61
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

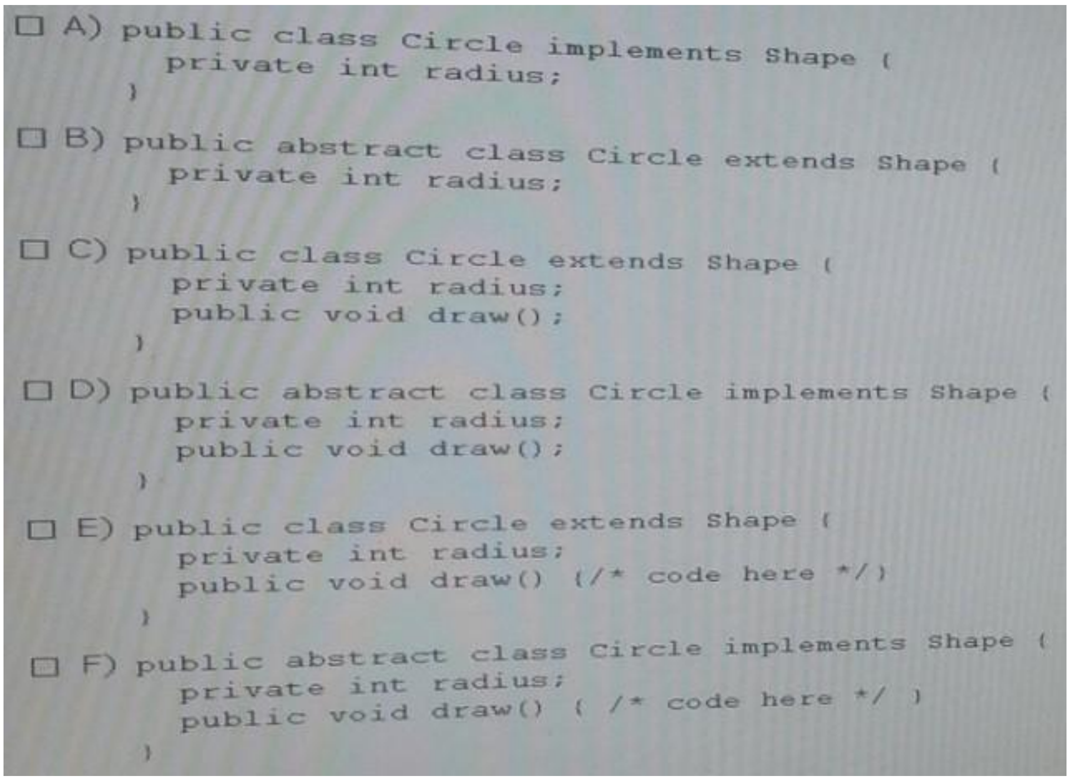
Answer: A, B, D

QUESTION: 62
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

Answer: B, E

Explanation:
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: 63
Given:
public class TestLoop1 {
public static void main(String[] args) {
int a = 0, z=10; while (a < z) { a++;

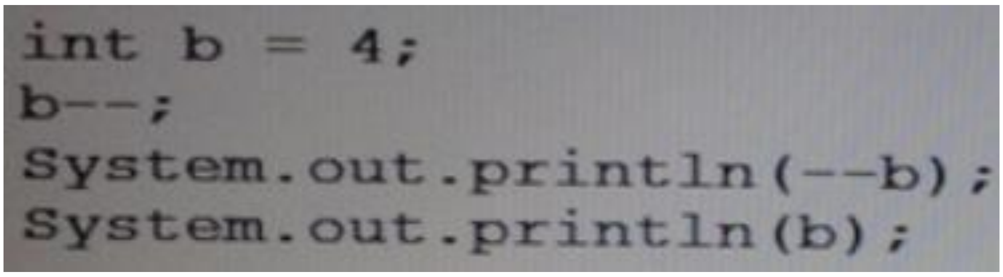
```
--z;
}
System.out.print(a + " : " + z);
}
}
```

What is the result?

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

Answer: A

QUESTION: 64
Given the code fragment:



```
int b = 4;
b--;
System.out.println(--b);
System.out.println(b);
```

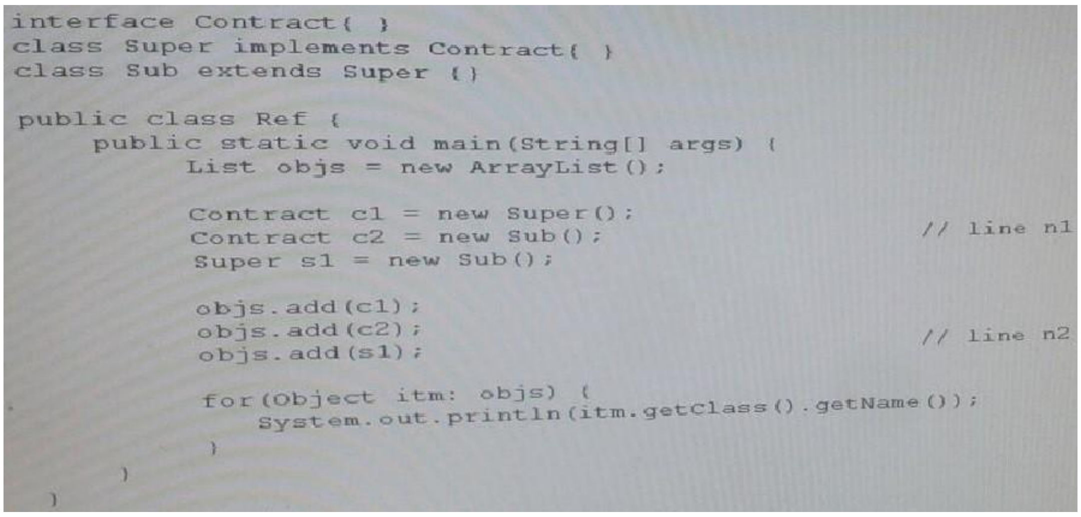
What is the result?

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

Answer: A

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

QUESTION: 65
Given the code fragment:

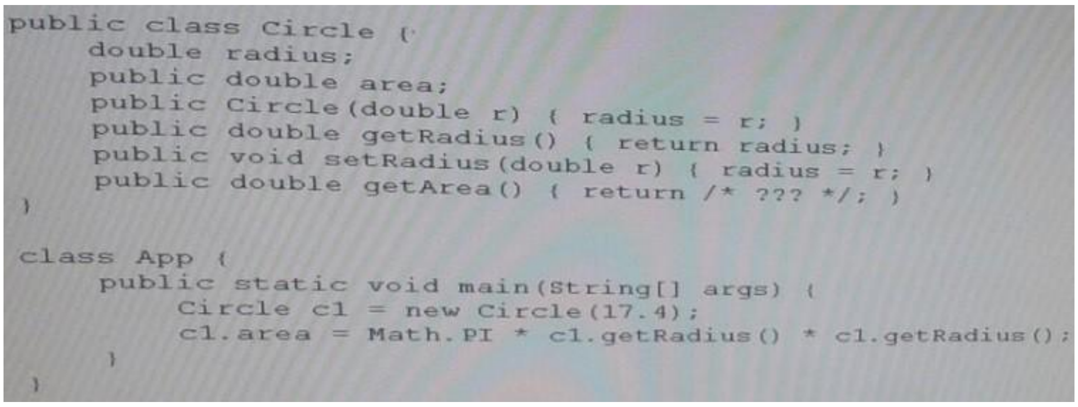


```
interface Contract{ }
class Super implements Contract{ }
class Sub extends Super {}

public class Ref {
    public static void main(String[] args) {
        List objs = new ArrayList();

        Contract c1 = new Super();
        Contract c2 = new Sub();                    // line n1
        Super s1 = new Sub();

        objs.add(c1);
        objs.add(c2);                               // line n2
        objs.add(s1);

        for(Object itm: objs) {
            System.out.println(itm.getClass().getName());
        }
    }
}
```

A. Super Sub Sub
B. Contract Contract Super
C. Compilation fails at line n1
D. Compilation fails at line n2

Answer: D

QUESTION: 66
Given:



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

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

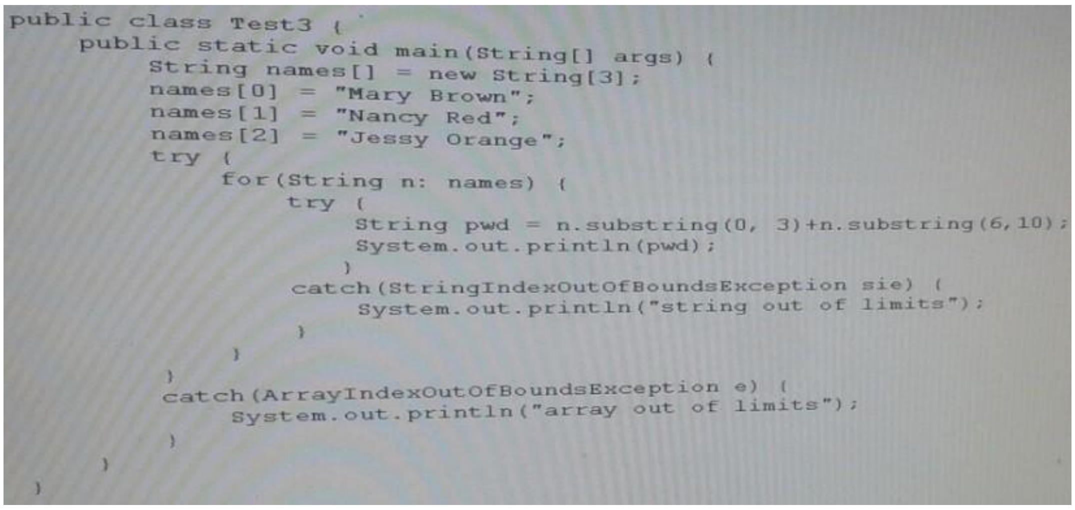
The class is poorly encapsulated. You need to change the circle class to compute and return the area instead. Which two modifications are necessary to ensure that the class is being properly encapsulated?

A. Remove the area field.
B. Change the getArea( ) method as follows:
public double getArea ( ) { return Match.PI * radius * radius; }
C. Add the following method:
public double getArea ( ) {area = Match.PI * radius * radius; }

D. Change the cacess modifier of the SerRadius ( ) method to be protected.

Answer: B, D

QUESTION: 67
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

Answer: A

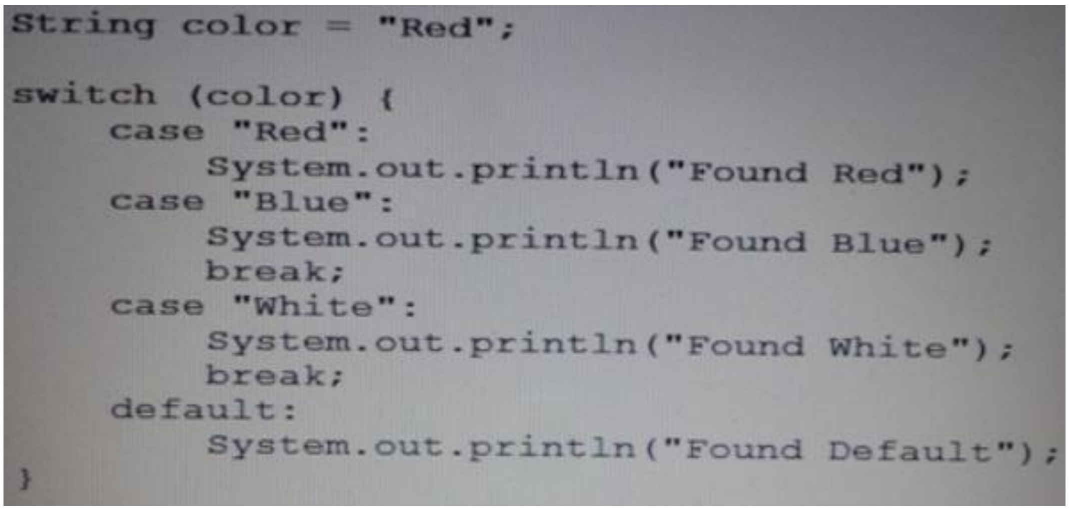
QUESTION: 68
Given:
public class TestField { int x;
int y;
public void doStuff(int x, int y) { this.x = x;
y =this.y;
}
public void display() { System.out.print(x + " " + y + " : ");
}
public static void main(String[] args) { TestField m1 = new TestField(); m1.x = 100;
m1.y = 200;
TestField m2 = new TestField(); m2.doStuff(m1.x, m1.y); m1.display();
m2.display();

}
}
What is the result?

A. 100 200 : 100 200
B. 100 0 : 100 0 :
C. 100 200 : 100 0 :
D. 100 0 : 100 200 :

Answer: C

QUESTION: 69
Given the code fragment:



```
String color = "Red";

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

What is the result?

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

Answer: B

Explanation:
As there is no break statement after the case "Red" statement the case Blue statement will run as well. Note: The body of a switch statement is known as a switch block. A statement in the switch block can be labeled with one or more case or default labels. The switch statement evaluates its expression, then executes all statements that follow the matching case label.
Each break statement terminates the enclosing switch statement. Control flow continues

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

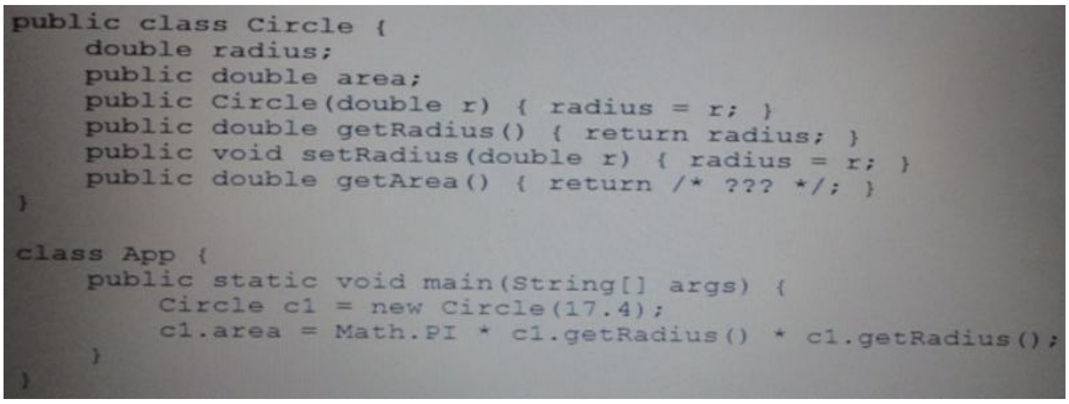
QUESTION: 70
Identify two benefits of using ArrayList over array in software development.

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

Answer: A, D

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: 71
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;}

Answer: B, C, D

QUESTION: 72
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 arra 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[]) {
D. toUpperCase();
}
}
E. for (int i:arra.length) { for (String x:arra) { arra[i].toUpperCase();
}
}

Answer: C

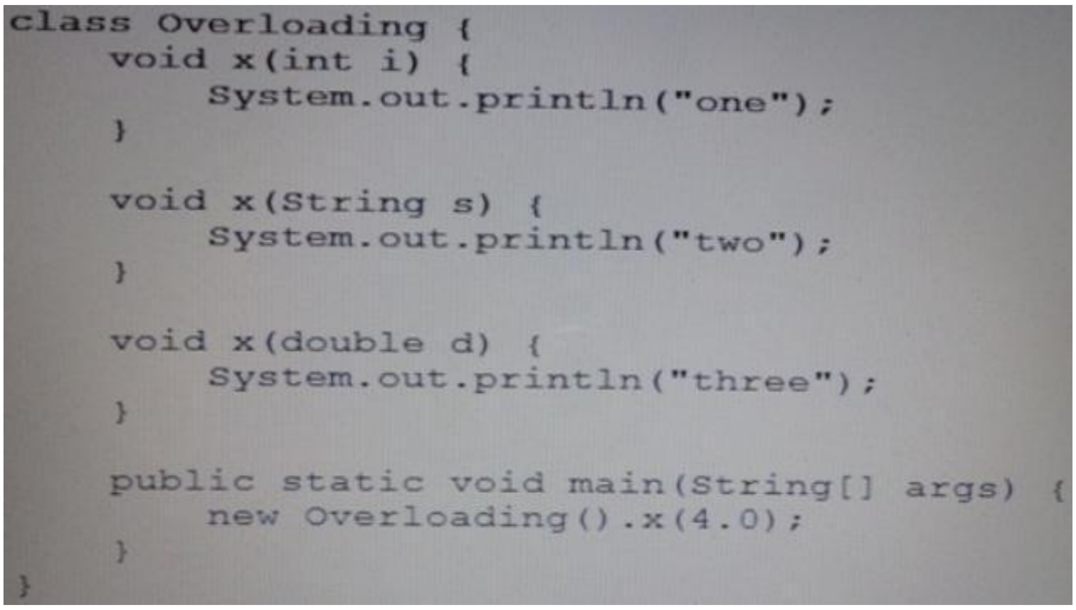
Explanation:
Incorrect:
not A: arra.length is 3, but the subarrays have 2, 3 and 4 elements. Index will be out of bound.
not B: The subarrys are of different lengths. Index will be out of bound. not D: Compile error.

QUESTION: 73
Which statement initializes a stringBuilder to a capacity of 128?

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

Answer: D

Explanation:
StringBuilder(int capacity) Constructs a string builder with no characters in it and an initial capacity specified by 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: 74
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

Answer: C

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.

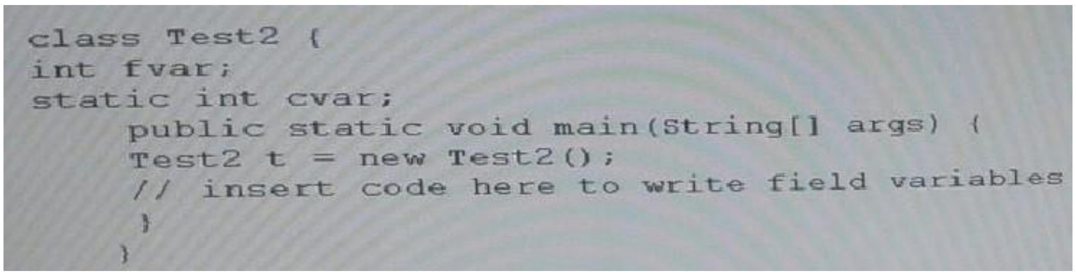
QUESTION: 75
Which two items can legally be contained within a java class declaration?

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

Answer: B, D

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

QUESTION: 76
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 code fragments, inserted independently, enable the code 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;

Answer: B

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)

Answer: C

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
Given the code fragment:
1. ArrayList<Integer> list = new ArrayList<>(1); 2. list.add(1001);
3. list.add(1002);

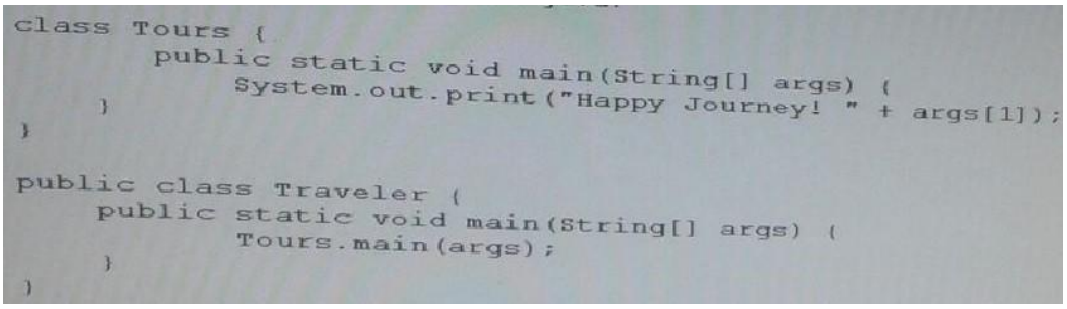
4. System.out.println(list.get(list.size())); What is the result?

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

Answer: C

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

QUESTION: 79
Given the code in a file Traveler.java:



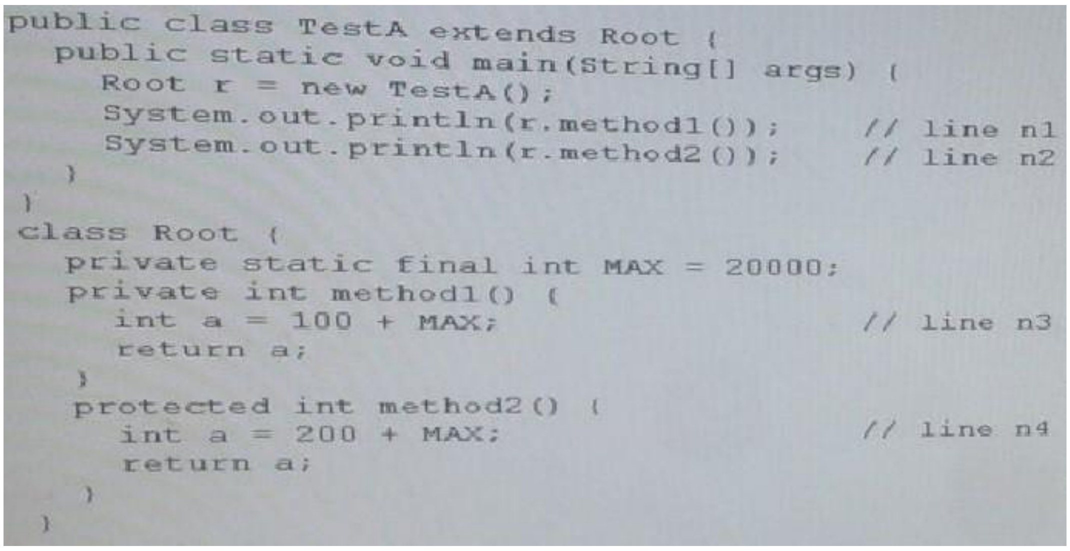
```
class Tours {
        public static void main(String[] args) {
              System.out.print("Happy Journey! " + args[1]);
     }
}

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

Answer: D

QUESTION: 80
Given the fragments:



```
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 causes a compilation error?

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

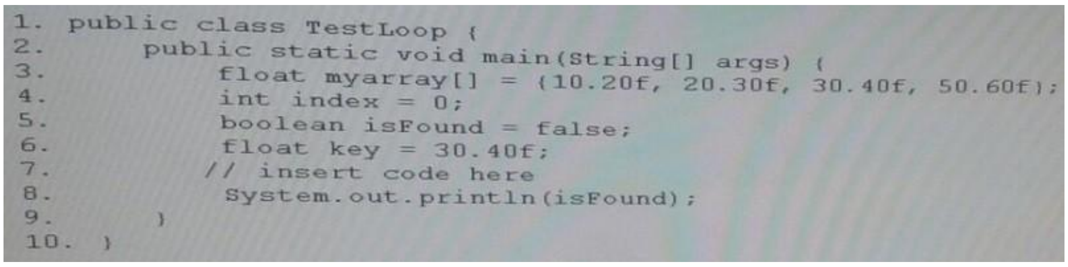
Answer: A

QUESTION: 81
A method is declared to take three arguments. A program calls this method and passes only two arguments. What is the results?

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 falsy value for its declared type.
F. An exception occurs when the method attempts to access the third argument.

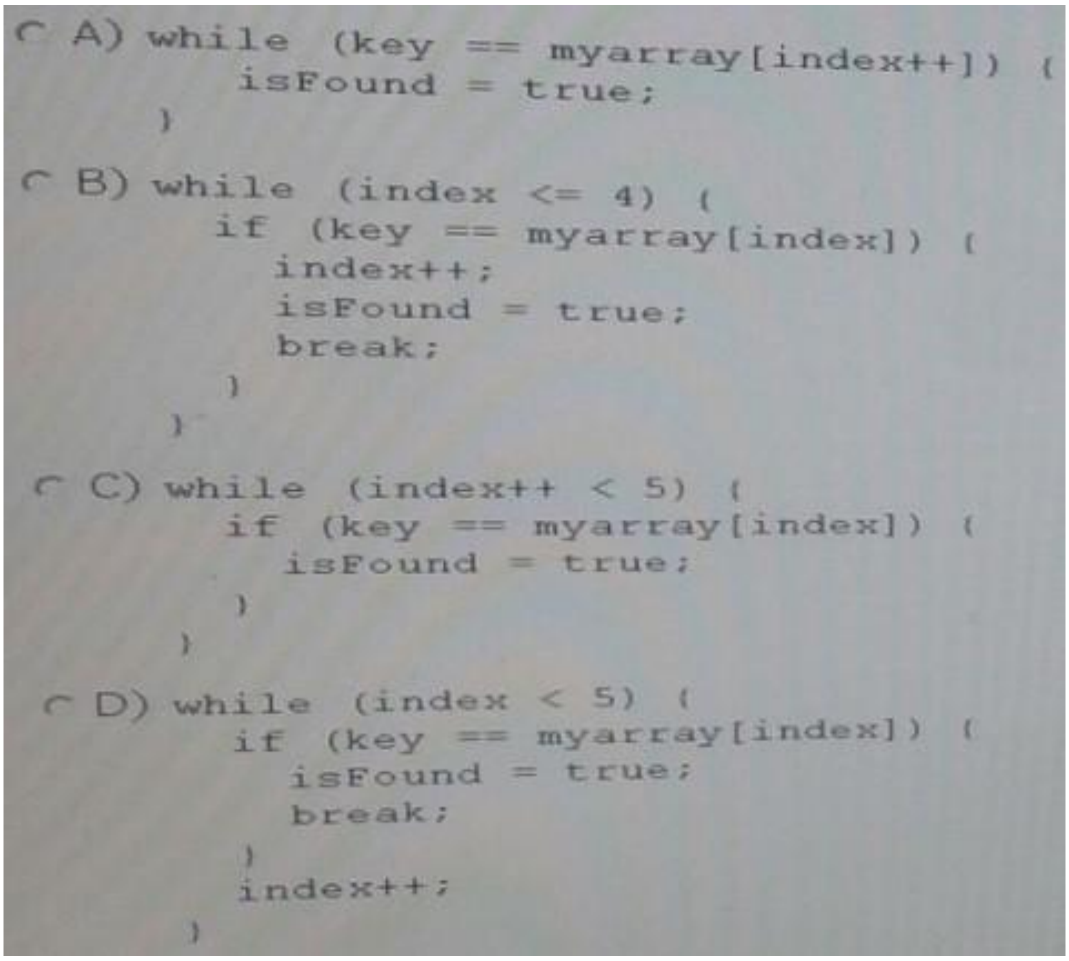
Answer: A

QUESTION: 82
Given:



```
1.  public class TestLoop {
2.      public static void main(String[] args) {
3.          float myarray[] = {10.20f, 20.30f, 30.40f, 50.60f};
4.          int index = 0;
5.          boolean isFound = false;
6.          float key = 30.40f;
7.          // insert code here
8.          System.out.println(isFound);
9.      }
10. }
```

Which code fragment, when inserted at line 7, enables the code print true?



```
○ A) while (key == myarray[index++]) {
         isFound = true;
     }

○ B) while (index <= 4) {
         if (key == myarray[index]) {
             index++;
             isFound = true;
             break;
         }
     }

○ C) while (index++ < 5) {
         if (key == myarray[index]) {
             isFound = true;
         }
     }

○ D) while (index < 5) {
         if (key == myarray[index]) {
             isFound = true;
             break;
         }
         index++;
     }
```

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

Answer: A

QUESTION: 83
Given the code fragment:

```
public class Test {
blic static void main(String[] args) { boolean isChecked = false;
int arry[] = {1,3,5,7,8,9};
int index = arry.length; while ( <code1> ) {
if (arry[index-1] % 2 ==0) { isChecked = true;
}
<code2>
}
System.out.print(arry(index]+", "+isChecked));
}
}
```

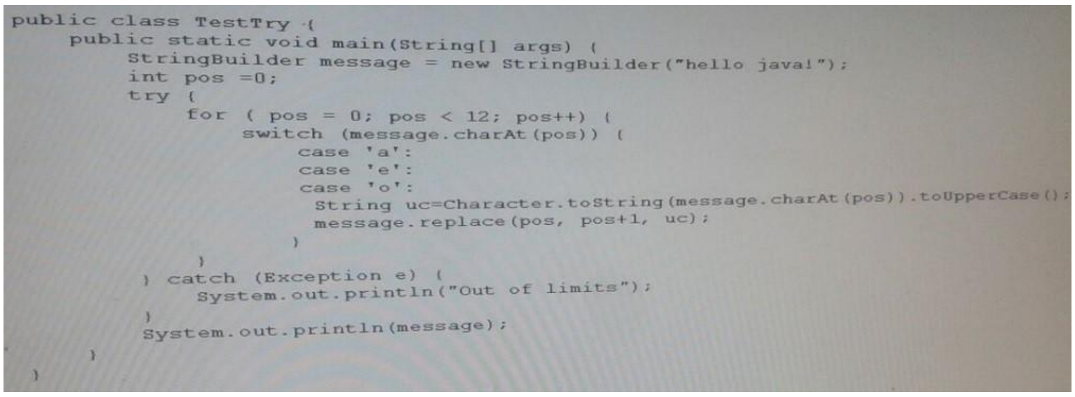
Which set of changes enable the code to print 1, true?

A. Replacing <code1> with index > 0 and replacing <code2> with index--;
B. Replacing <code1> with index > 0 and replacing <code2> with --index;
C. Replacing <code1> with index > 5 and replacing <code2> with --index ;
D. Replacing <code1> with index and replacing <code2> with --index ;

Answer: A

Explanation:
Note: Code in B (code2 is --index;). also works fine.

QUESTION: 84
Given:



```
public class TestTry {
    public static void main(String[] args) {
        StringBuilder message = new StringBuilder("hello java!");
        int pos =0;
        try {
            for ( pos = 0; pos < 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

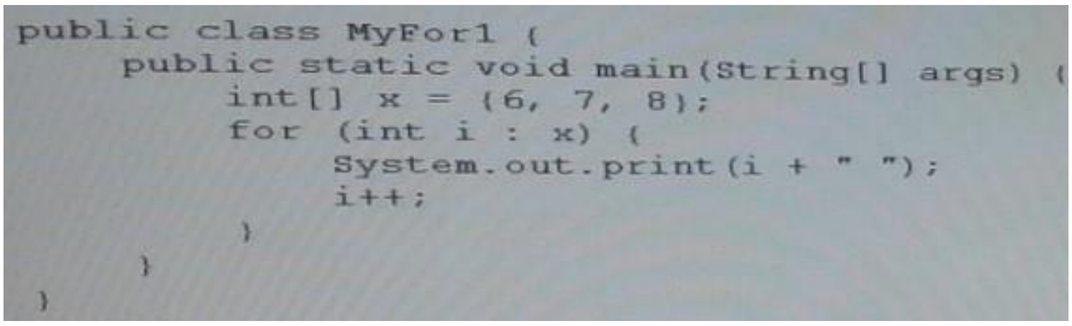
Answer: C

QUESTION: 85
View the exhibit:**Missing 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();

Answer: C

QUESTION: 86
Given:



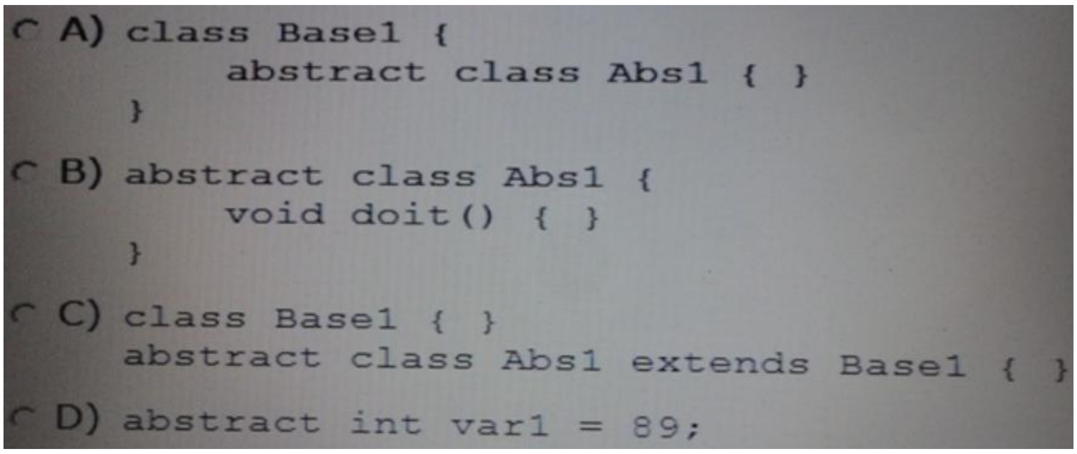
```
public class MyFor1 {
    public static void main(String[] args) {
        int[] x = {6, 7, 8};
        for (int i : x) {
            System.out.print(i + " ");
            i++;
        }
    }
}
```

What is the result?

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

Answer: A

QUESTION: 87
Which code fragment is illegal?



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

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

Answer: D

Explanation:
The abstract keyword cannot be used to declare an int variable. The abstract keyword is used to declare a class or method to 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: 88
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.

Answer: B, C

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: 89
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;}

Answer: A, D

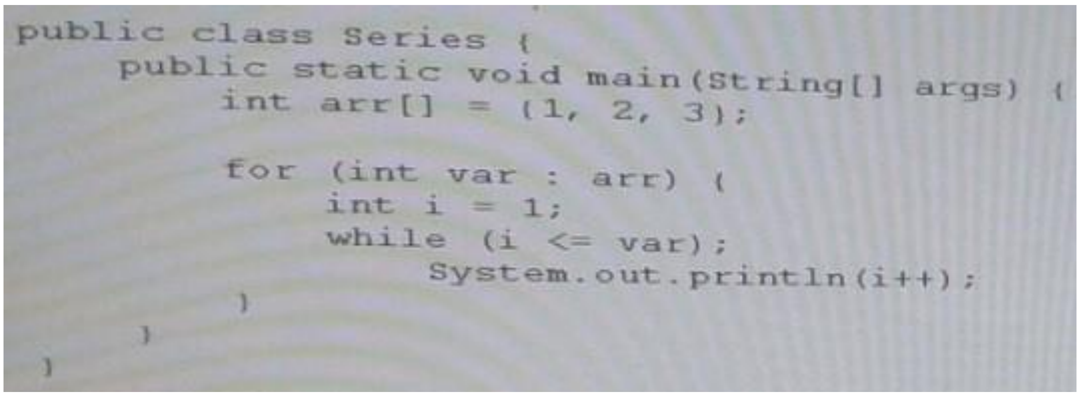
Explanation:
Incorrect:
Not B: illegal combination of modifiers: protected and public not C: setNum method cannot be private.
not E: getNum method cannot be private.

QUESTION: 90
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();

Answer: C

QUESTION: 91
Given:



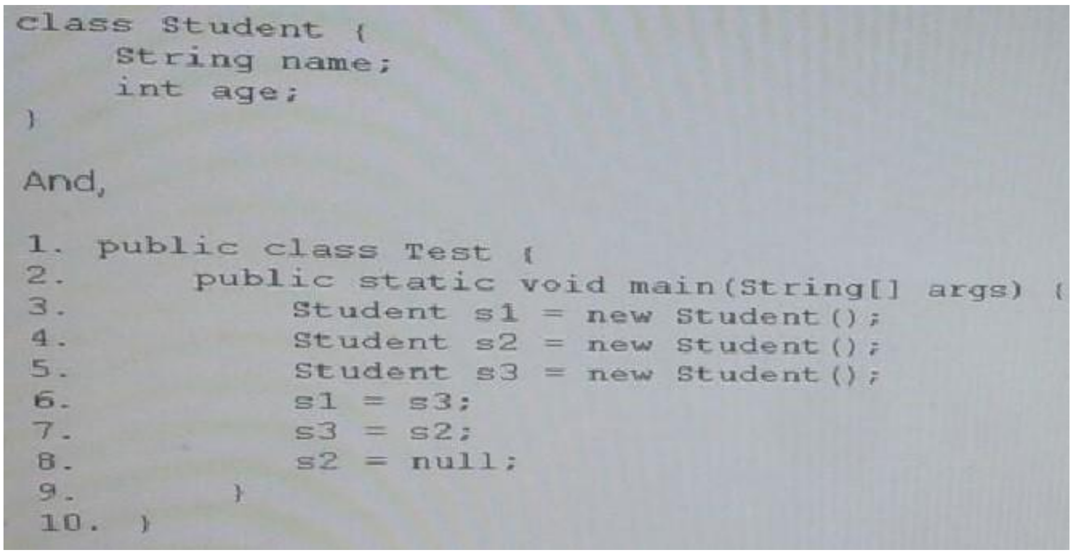
```
public class Series {
    public static void main(String[] args) {
        int arr[] = {1, 2, 3};

        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

Answer: E

QUESTION: 92
Given the code fragment:



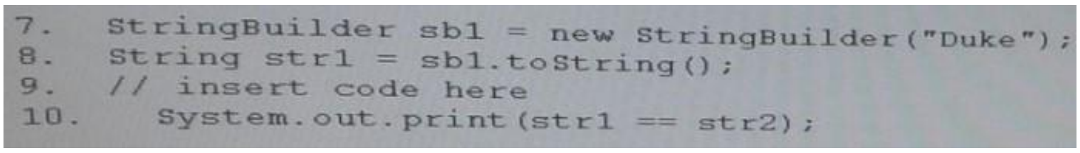
```
class Student {
     String name;
     int age;
}

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

Answer: C

QUESTION: 93
Given:



```
7.   StringBuilder sb1 = new StringBuilder("Duke");
8.   String str1 = sb1.toString();
9.   // insert code here
10.    System.out.print(str1 == str2);
```

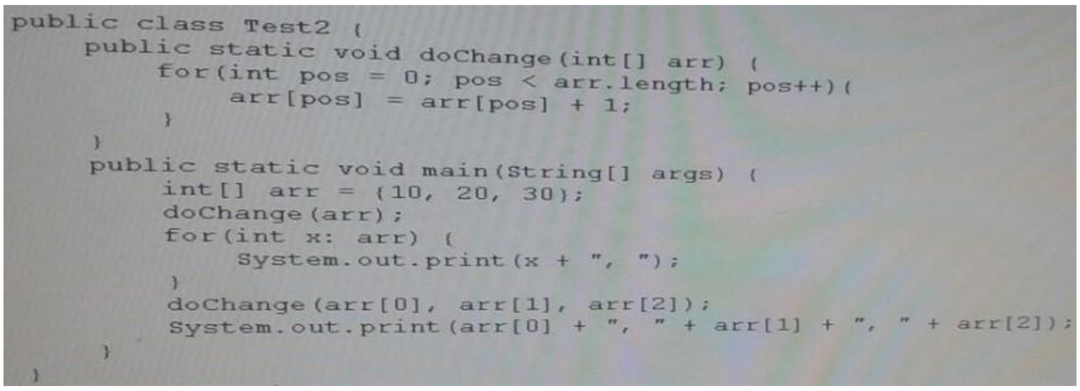
Which code fragment, when inserted at line 9, enables the code to print true?

A. String str2 = str1;
B. String str2 = new string (str1);
C. String str2 = sb1.toString();
D. String str2 = “Duke”;

Answer: B

QUESTION: 94

Given:



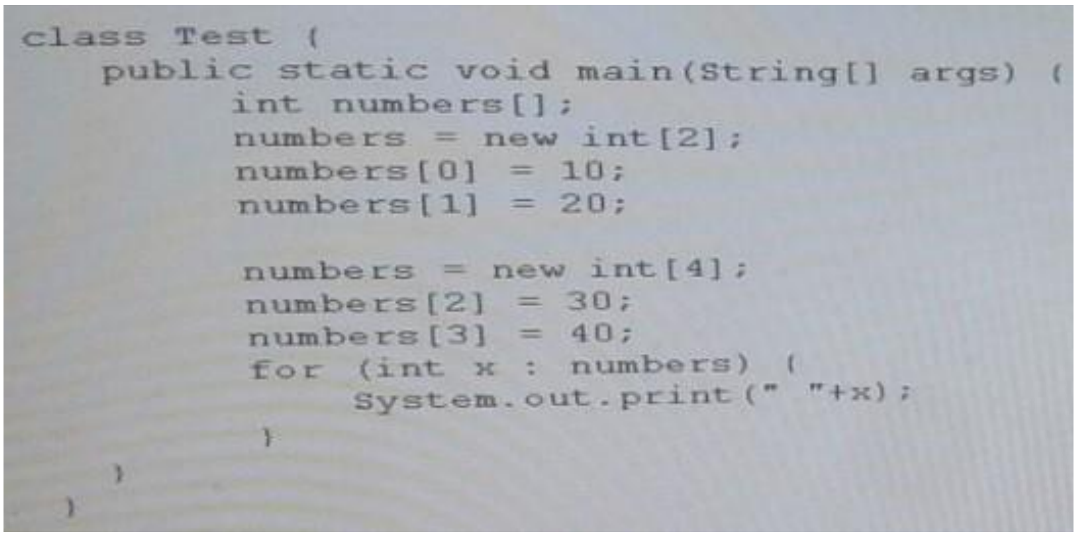
```
public class Test2 {
    public static void doChange(int[] arr) {
        for(int pos = 0; pos < arr.length; pos++){
            arr[pos] = arr[pos] + 1;
        }
    }
    public static void main(String[] args) {
        int[] arr = {10, 20, 30};
        doChange(arr);
        for(int x: arr) {
            System.out.print(x + ", ");
        }
        doChange(arr[0], arr[1], arr[2]);
        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

Answer: D

QUESTION: 95
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

Answer: A

QUESTION: 96

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

Answer: B

Explanation:
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: 97

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;}}
```

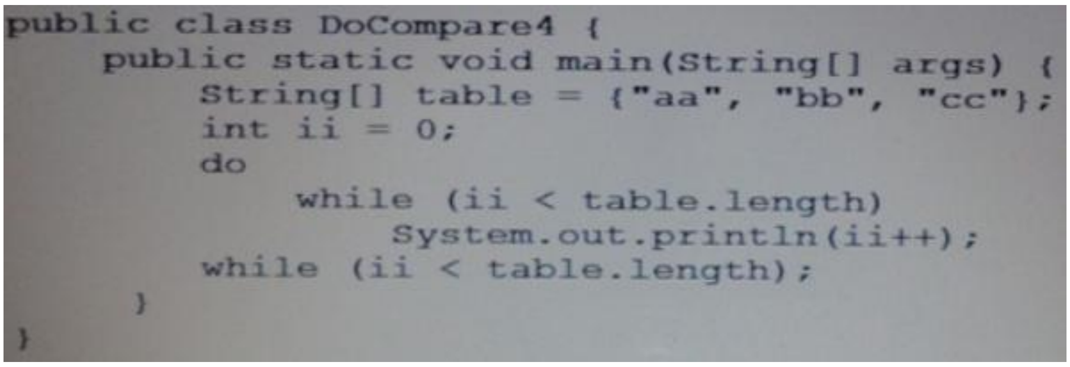
This class needs to protect an invariant on the sum field.
Which three members must have the private access modifier to ensure that this invariant is maintained?

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

Answer: C, E, F

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

QUESTION: 98
Given:



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

What is the result?

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

Answer: B

Explanation:

table.length is 3. So the do-while loop will run 3 times with ii=0, ii=1 and ii=2. The second while statement will break the do-loop when ii = 3.
Note: The Java programming language provides a do-while statement, which can be expressed as follows:
do { statement(s)
} while (expression);

QUESTION: 99
Given:
interface Pet { }
class Dog implements Pet { } public 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();

Answer: A, D, E

Explanation:
Incorrect:
Not B, not C: Pet is abstact, cannot be instantiated. Not F: incompatible type. Required Beagle, found Dog.

QUESTION: 100
Given:
public class Test { static boolean bVar;
public static void main(String[] args) { boolean bVar1 = true;
int count =8; do {
System.out.println("Hello Java! " +count); if (count >= 7) {
bVar1 = false;
}
} while (bVar != bVar1 && count > 4); count -= 2;
}
}
What is the result?

A. Hello Java! 8 Hello Java! 6 Hello Java! 4
B. Hello Java! 8 Hello Java! 6
C. Hello Java! 8
D. Compilation fails

Answer: C

Explanation:
Hello Java! 8