Oracle

Exam 1z0-803

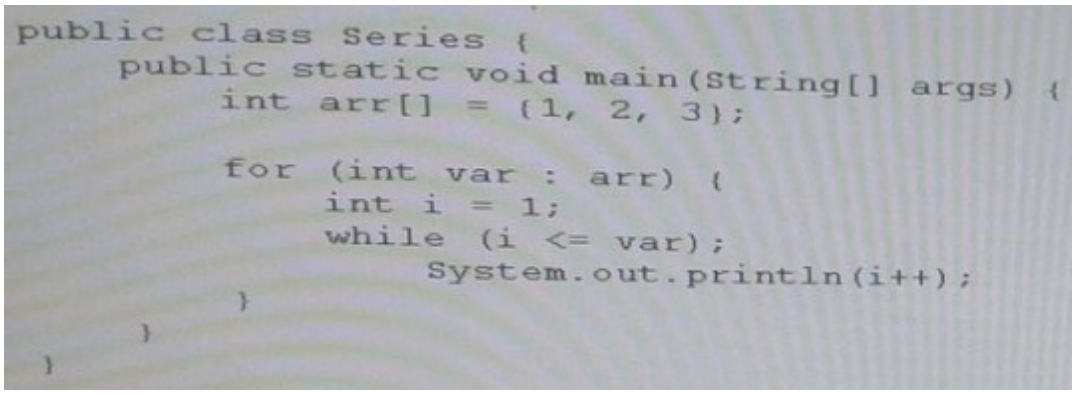
Java SE 7 Programmer I

Version: 11.5

[Total Questions: 216]

Question No : 1

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 No : 2

The catch clause argument is always of type_____.

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

Answer: C

Explanation:

Because all exceptions in Java are the sub-class ofjava.lang.Exceptionclass, 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 canhandle 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-specificException 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 forby 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 No : 3

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 No : 4

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

Answer: A

Question No : 5

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.

Answer: A

Question No : 6

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());
}
}
```

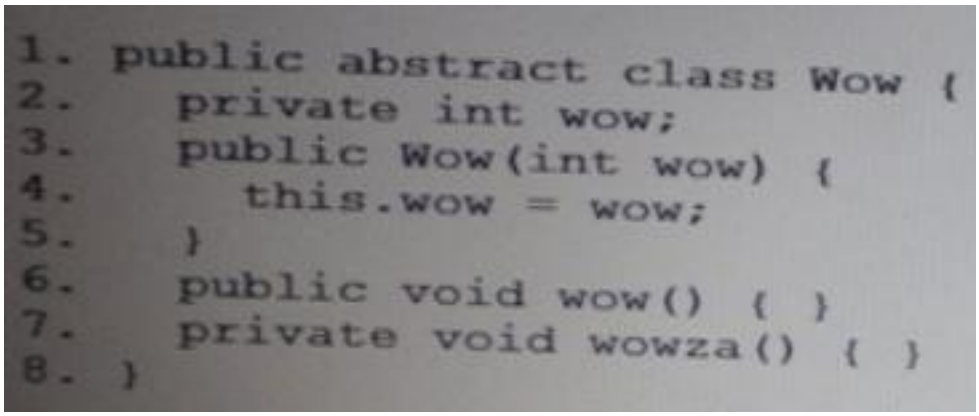
What is the result?

A. : Fresco
B. null : Fresco
C. Fresco : Fresco
D. A NullPointerException is thrown at runtime

Answer: B

Question No : 7

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.

Answer: A

Question No : 8

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?

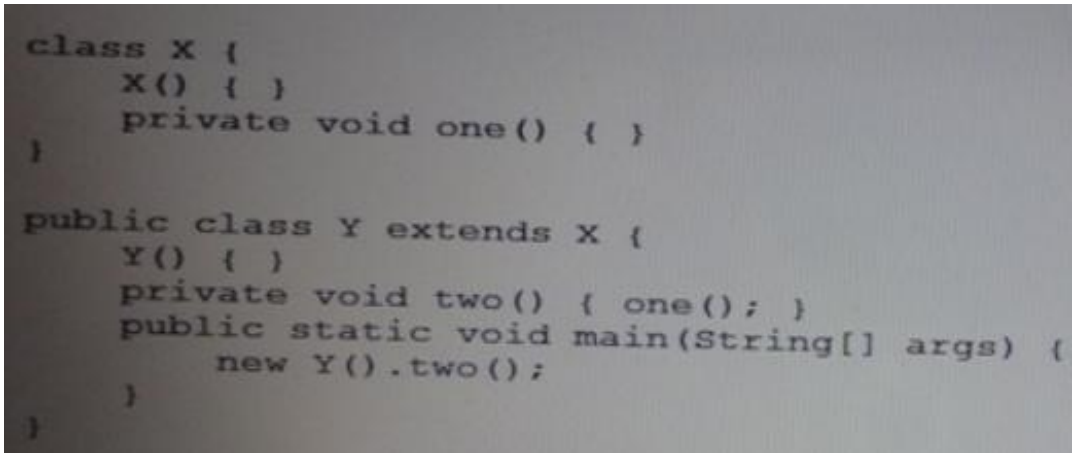
A. true, false
B. false, true
C. true, true
D. false, false

Answer: B

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

Question No : 9

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 No : 10

Given the code fragment:

float x = 22.00f % 3.00f;

int y = 22 % 3;

System.out.print(x + ", "+ y);

What is the result?

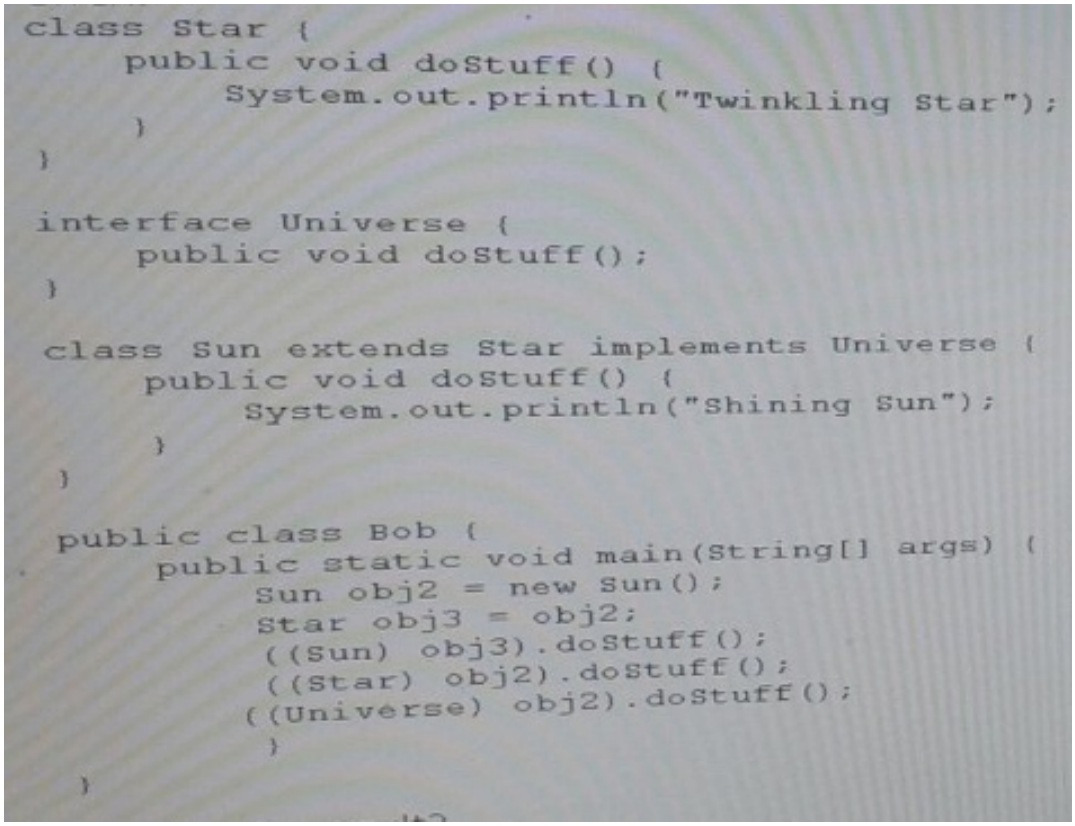
A. 1.0, 1
B. 1.0f, 1

C. 7.33, 7
D. Compilation fails
E. An exception is thrown at runtime

Answer: A

Question No : 11

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 No : 12

Given the code fragment:

Stringh1 = "Bob";

String h2 = new String ("Bob");

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

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

Answer: B

Explanation:
The equals method compares values for equality.

Question No : 13

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 No : 14

Given:

```
public class FieldInit {
char c;
booleanb;
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();
}
}
```

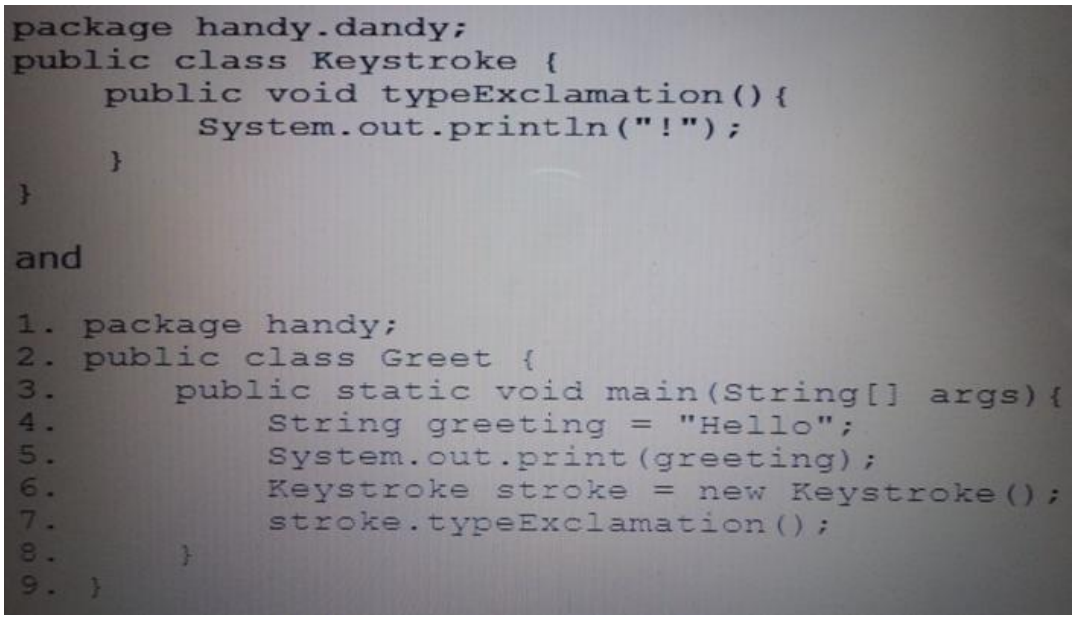
What is the result?

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

Answer: D

Question No : 15

Given:



```
package handy.dandy;
public class Keystroke {
    public void typeExclamation(){
        System.out.println("!");
    }
}
```

and

```
1. package handy;
2. public class Greet {
3.     public static void main(String[] args){
4.         String greeting = "Hello";
5.         System.out.print(greeting);
6.         Keystroke stroke = new Keystroke();
7.         stroke.typeExclamation();
8.     }
9. }
```

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

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

Answer: C,E,F

Explanation:

Three separate solutions:

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

D: We can import the hold dandy class

F: we can import the specific method

Question No : 16

```
public class ForTest {

public static void main(String[] args) {

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

for ( foo ) {

}

}

}
```

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

A. int i: array
B. int i = 0; i < 1; i++
C. ;;
D. ; i < 1; i++
E. ; i < 1;

Answer: A,B,C

Question No : 17

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 No : 18

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 No : 19

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

Answer: A

Explanation: 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 No : 20

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 programto 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 anIOException and ensuring that the program can recover if one occurs

Answer: A,B,D

Question No : 21

Given:

```
class Patient {
    String name;
    public Patient(String name) {
        this.name = name;
    }
}
```

And the code fragment:

```
 8. public class Test {
 9.     public static void main(String[] args) {
10.         List ps = new ArrayList();
11.         Patient p2 = new Patient("Mike");
12.         ps.add(p2);
13.
14.         // insert code here
15.
16.         if (f >=0 ) {
17.             System.out.print("Mike Found");
18.         }
19.     }
20. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

A. int f = ps.indexOf {new patient (“Mike”)};
B. int f = ps.indexOf (patient(“Mike”));
C. patient p = new Patient (“Mike”);
int f = pas.indexOf(P)
D. int f = ps.indexOf(p2);

Answer: C

Question No : 22

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 No : 23

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 No : 24

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 No : 25

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(intnum) {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 No : 26

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 No : 27

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?

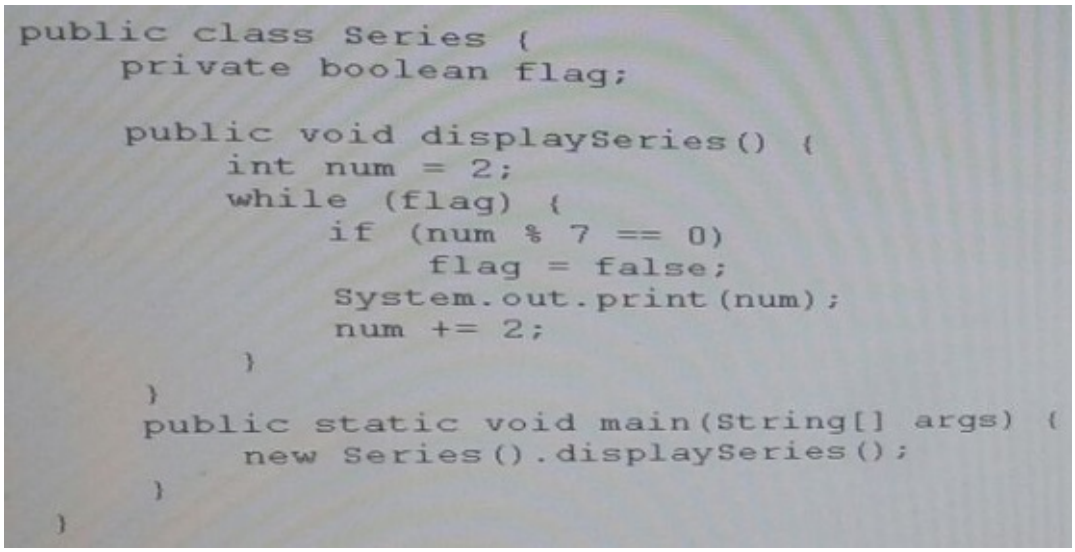
A. log3:false
log4:true
B. log3:true
log4:true
C. log3:true
log4:false

D. log3:false
log4:false

Answer: B

Question No : 28

Given:



```
public class Series {
    private boolean flag;

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

What is the result?

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

Answer: B

Question No : 29

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

Answer: A

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

Question No : 30

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

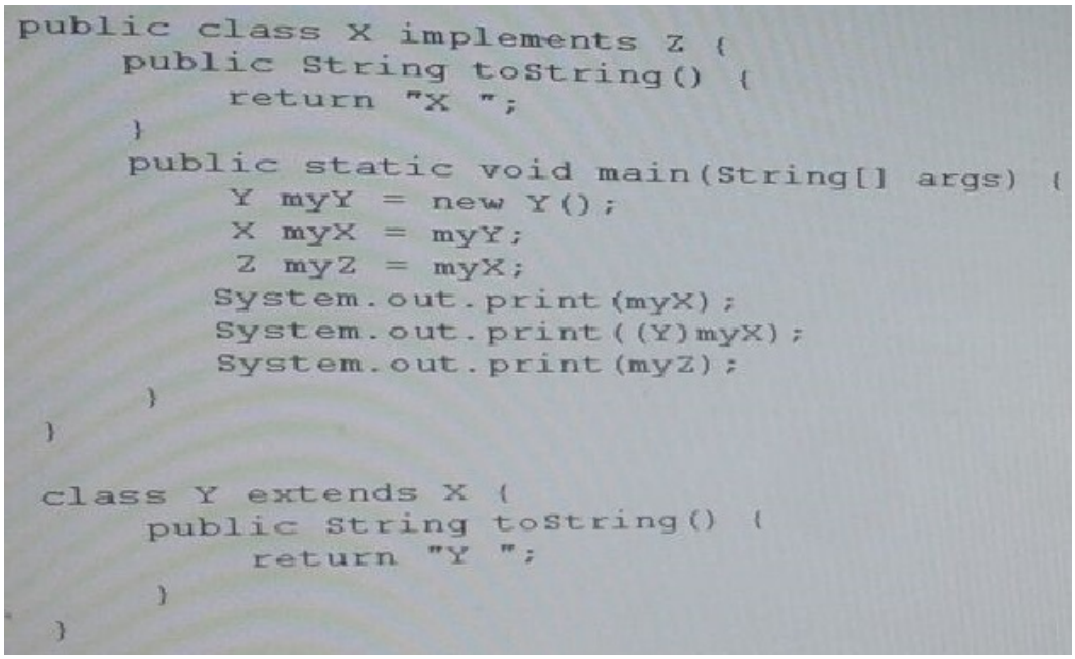
A. It can take arguments.
B. It has private access modifier in its declaration.
C. It can be overloaded.
D. The default constructor of a subclass always invokes the no-argument constructor of its superclass.

Answer: D

Explanation: 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 No : 31

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

Answer: D

Question No : 32

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. Anexception is thrown at run time due to error on line 4
D. 1002

Answer: C

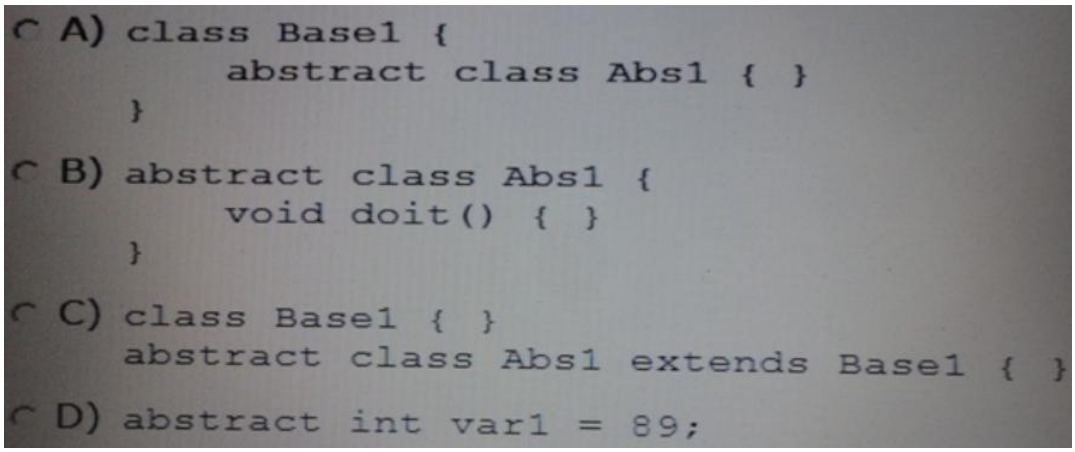
Explanation:

The code compiles fine.

At runtime an IndexOutOfBoundsException is thrown when the second list item is added.

Question No : 33

Which code fragment is illegal?



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 beabstract[3]. An abstract method has no implementation; all classes containing abstract methods must themselves be abstract, although not all abstract classes have abstract methods.

Question No : 34

Given:

```
public static void main(String[] args){
        int a, b, c = 0;
        int a, b, c;
        int g, int h, int i = 0;
        int d, e, F;
        Int k, l, m = 0;
}
```

Which two declarations will compile?

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

Answer: A,D

Question No : 35

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 No : 36

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 No : 37

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")

Answer: D
Explanation:
Equals works fine on strings equals produces a Boolean value.

Question No : 38

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);

}

}
```

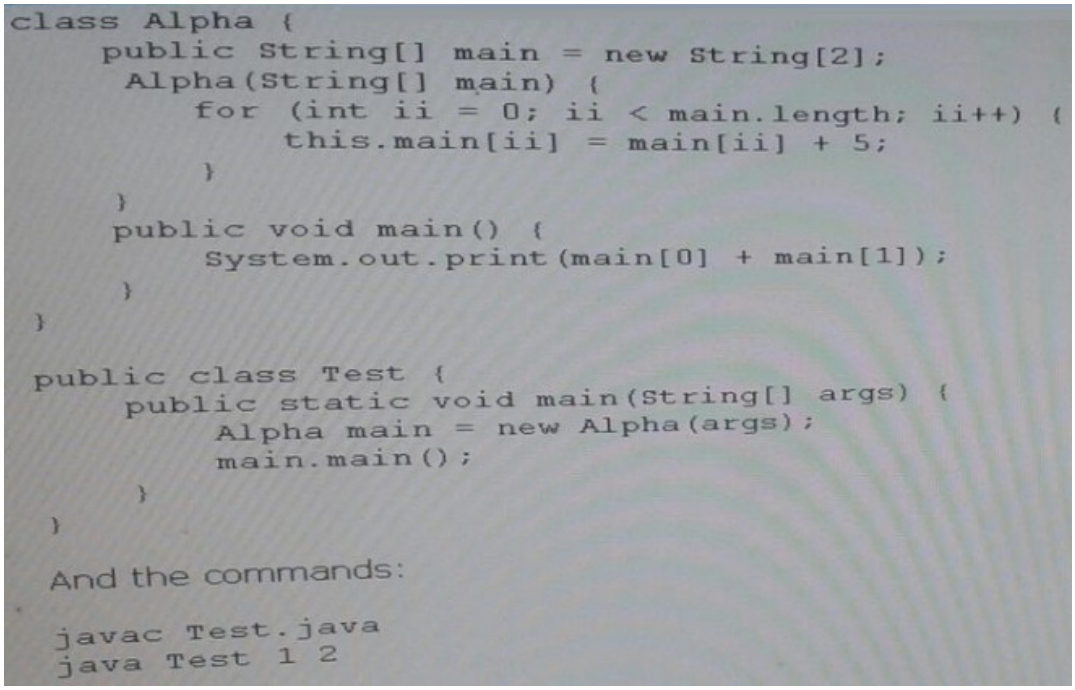
What is the result?

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

Answer: A

Question No : 39

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

Answer: D

Question No : 40

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 isexcuted?

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

Answer: C

Question No : 41

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 No : 42

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 thepossible errors
C. Improves the program structure because the programmer can choose where to handle exceptions
D. Improves the program structure because exceptions must be handled in the method in which they occurred
E. allows the creation of new exceptions that are tailored to the particular program being

Answer: A,C,E

Explanation:

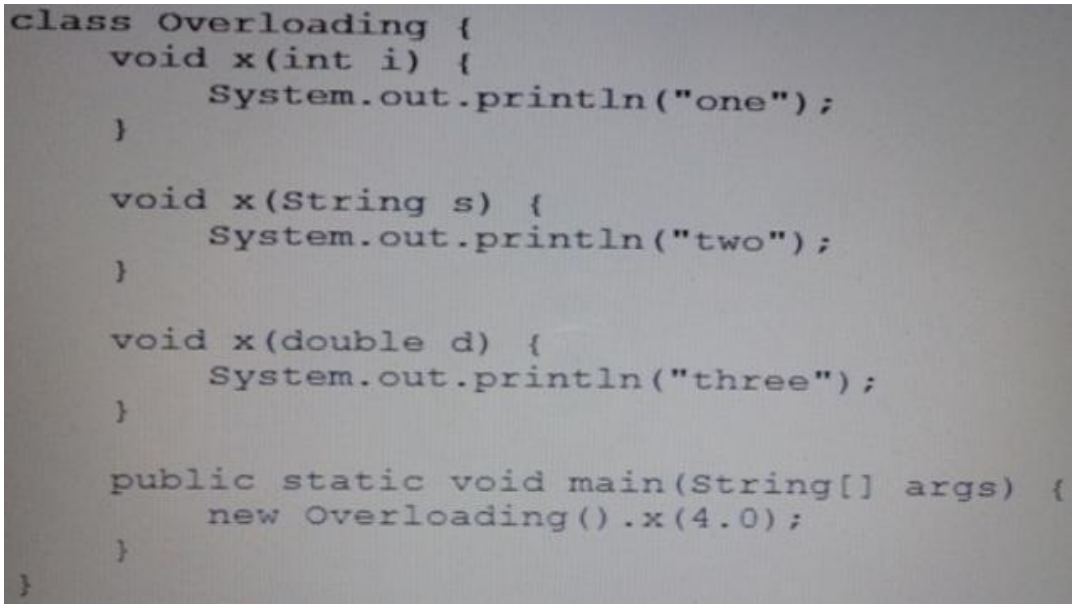
A: The error handling is separated from the normal program logic.

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

E: You can create your own exceptions.

Question No : 43

Given:



```
class Overloading {
    void x(int i) {
        System.out.println("one");
    }

    void x(String s) {
        System.out.println("two");
    }

    void x(double d) {
        System.out.println("three");
    }

    public static void main(String[] args) {
        new Overloading().x(4.0);
    }
}
```

What is the result?

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

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 No : 44

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");

Answer: D

Question No : 45

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:

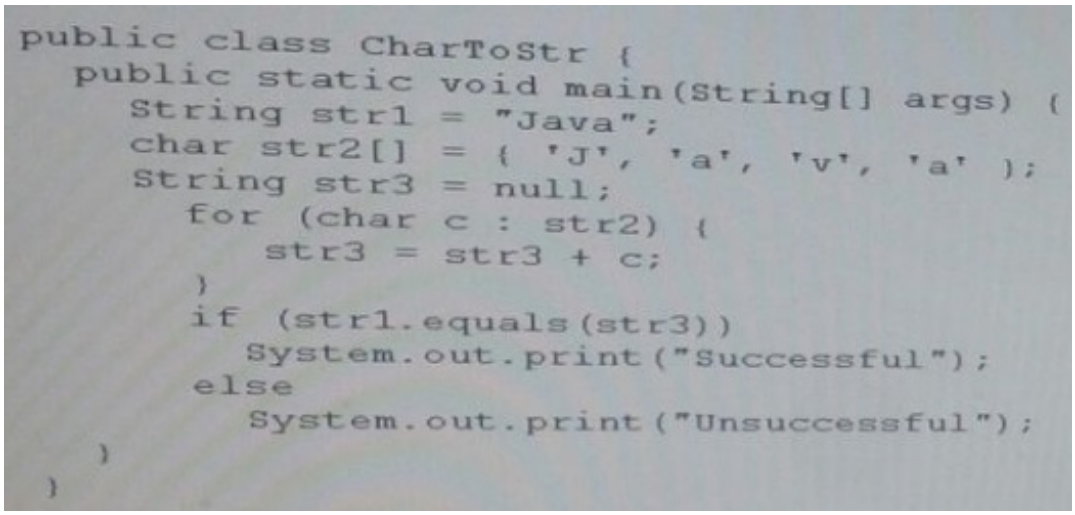
```
while (expression) {
statement(s)
}
```

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 No : 46

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

Answer: C

Question No : 47

TION NO: 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 inthread "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;

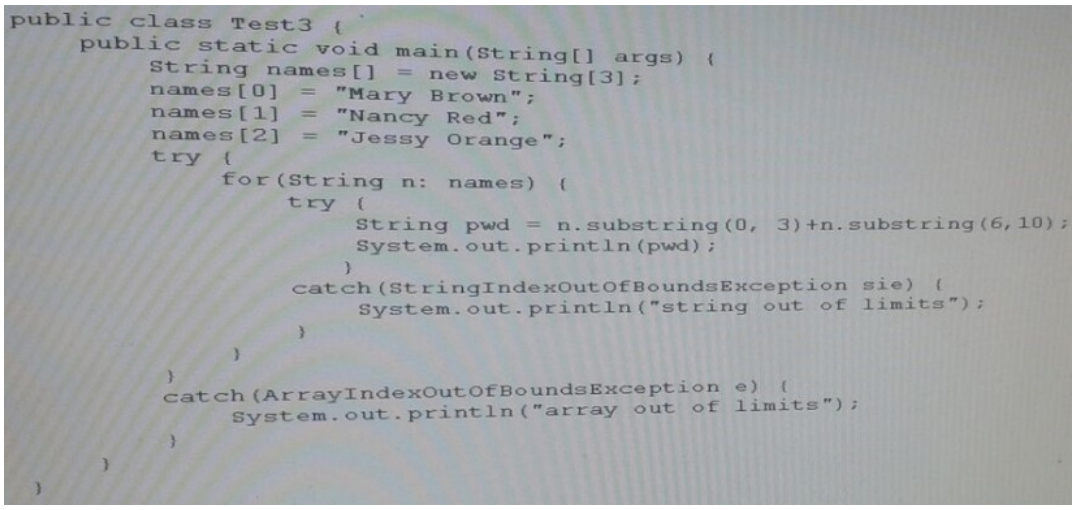
A runtime exception is thrown as anArrayIndexOutOfBoundsException.

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 errorsor improper use of an API.

Runtime exceptions*are not subject*to the Catch or Specify Requirement. Runtime exceptions are those indicated byRuntimeExceptionand its subclasses.

Question No : 48

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 No : 49

Given the for loop construct:

for (expr1 ; expr2 ; expr3) {

statement;

}

Which two statements aretrue?

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 No : 50

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

Answer: D

Question No : 51

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 isthe 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

Question No : 52

Given:

```
1.  public class Speak {
2.      public static void main(String[] args){
3.          Speak speakIt = new Tell();
4.          Tell tellIt = new Tell();
5.          speakIt.tellItLikeItIs();
6.          (Truth)speakIt.tellItLikeItIs();
7.          ((Truth)speakIt).tellItLikeItIs();
8.          tellIt.tellItLikeItIs();
9.          (Truth)tellIt.tellItLikeItIs();
10.         ((Truth)tellIt).tellItLikeItIs();
11.     }
12. }
13. class Tell extends Speak implements Truth {
14.     public void tellItLikeItIs() {
15.         System.out.println("Right on!");
16.     }
17. }
18. interface Truth { public void tellItLikeItIs(); }
```

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

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

Answer: C,D,F

Question No : 53

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?

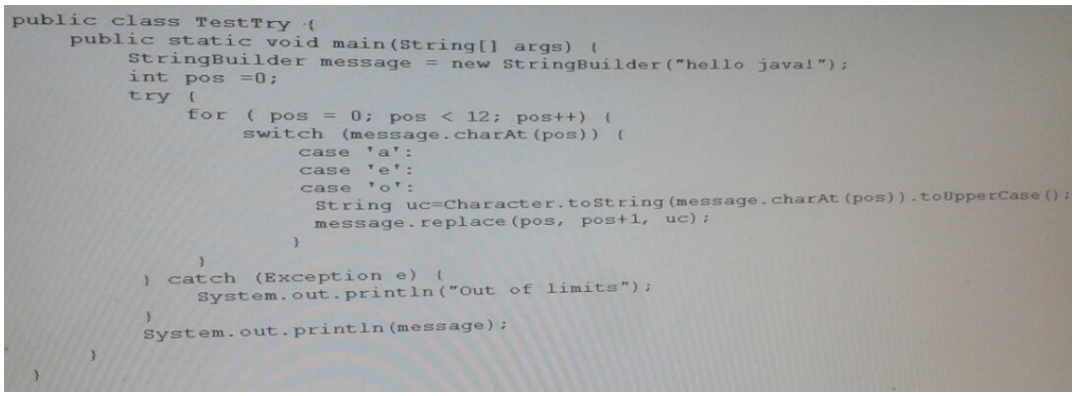
A. 13480.0
B. 13480.02
C. Compilation fails

D. An exception is thrown at runtime

Answer: A

Question No : 54

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);
    }
}
```

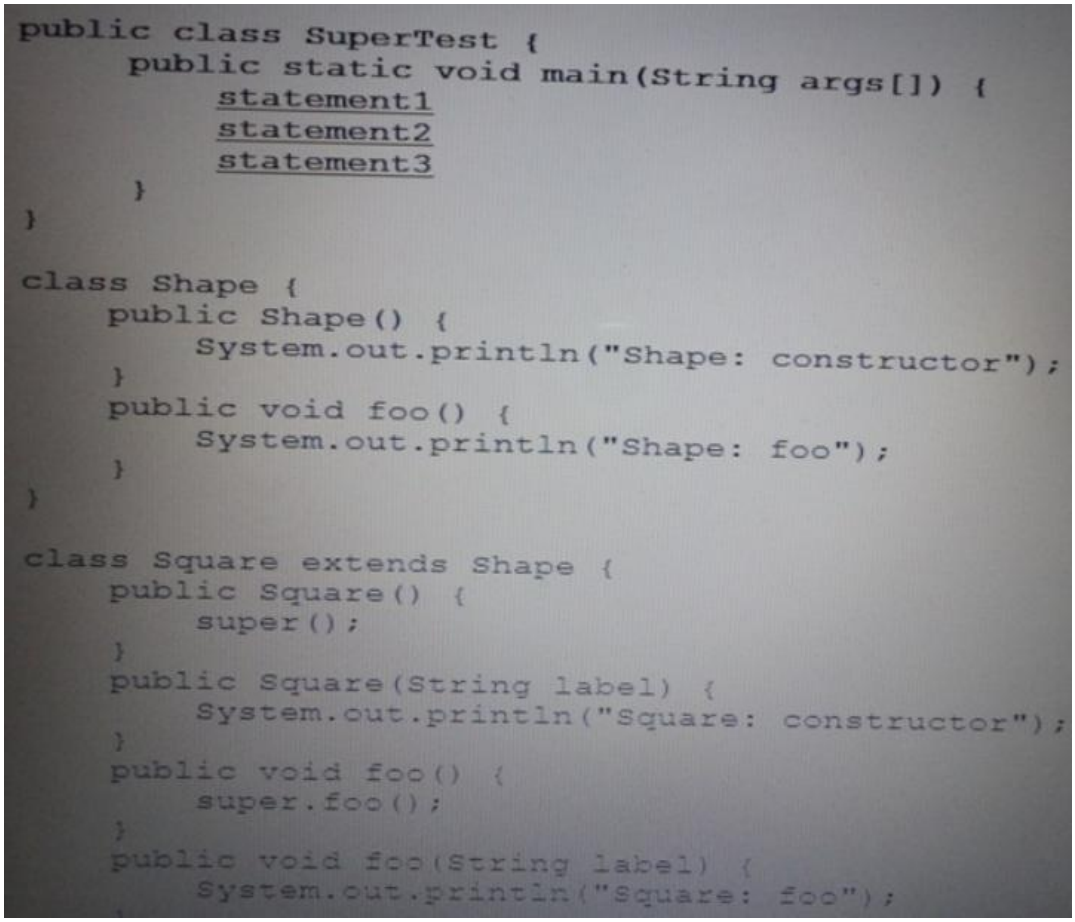
What is the result?

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

Answer: C

Question No : 55

Given:



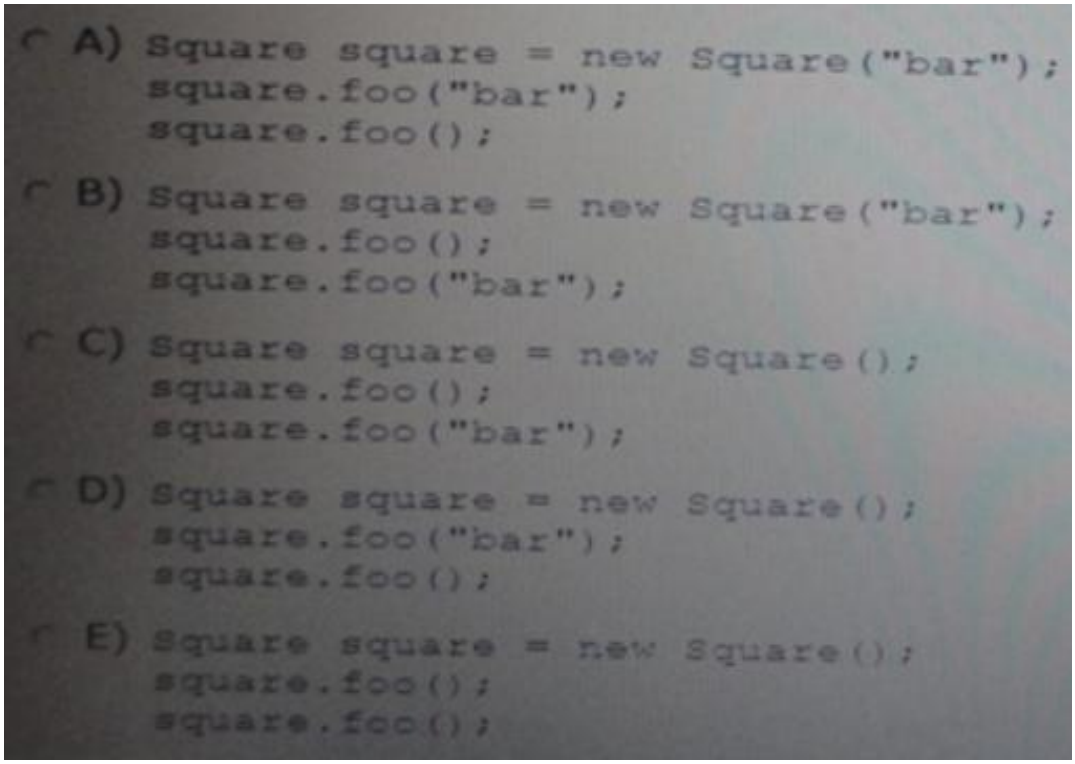
```
public class SuperTest {
     public static void main(String args[]) {
          statement1
          statement2
          statement3
     }
}

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 shouldstatement1,statement2, andstatement3, be respectively, in order to produce the result?

Shape: constructor

Square: foo

Shape: foo

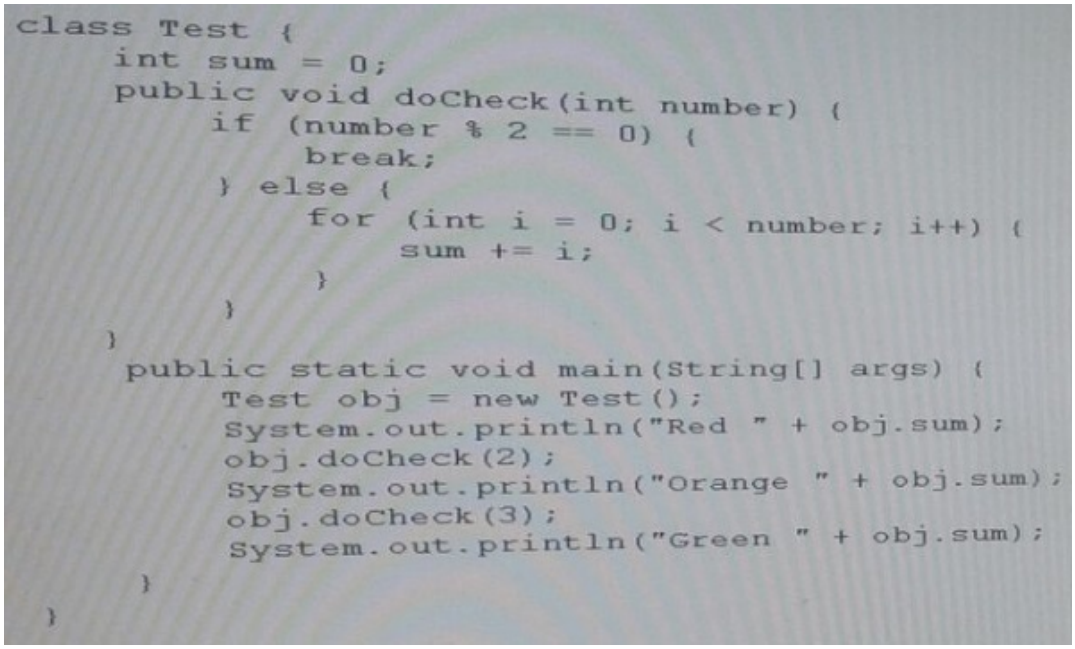


A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: D

Question No : 56

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

Answer: E

Question No : 57

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 No : 58

Given:

```
public class App {

// Insert code here

System.out.print("Welcome to the world of Java");

}

}
```

Which two code fragments, when inserted independently at line // Insert code here, enable the program to execute and print the welcome message on the screen?

A. static public void main (String [] args) {
B. static void main (String [] args) {
C. public static void Main (String [] args) {
D. public staticvoid main (String [] args) {

E. public void main (String [] args) {

Answer: A,D

Explanation:

Incorrect:

Not B: No main class found.

Not C: Main method not found

not E: Main method is not static.

Question No : 59

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 ….

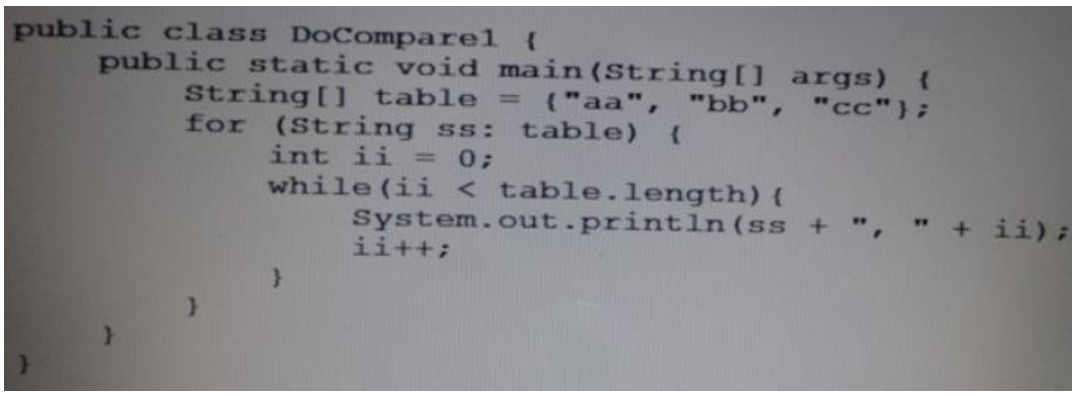
Answer: B

Explanation:

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

Question No : 60

Given:



```
public class DoCompare1 {
    public static void main(String[] args) {
        String[] table = {"aa", "bb", "cc"};
        for (String ss: table) {
            int ii = 0;
            while(ii < table.length){
                System.out.println(ss + ", " + ii);
                ii++;
            }
        }
    }
}
```

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

A. Zero
B. Once
C. Twice
D. Thrice
E. Compilation fails.

Answer: A

Question No : 61

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

Answer: C

Question No : 62

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?

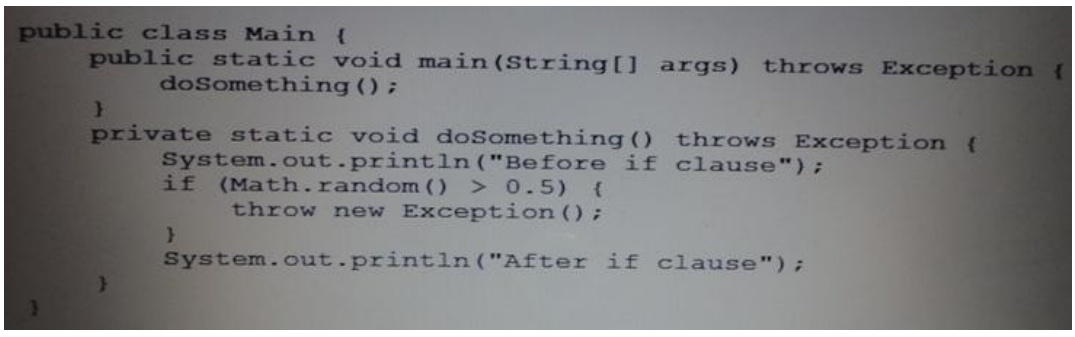
A. 8
B. 9
C. 11
D. 10
E. Compilation fails

Answer: B

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

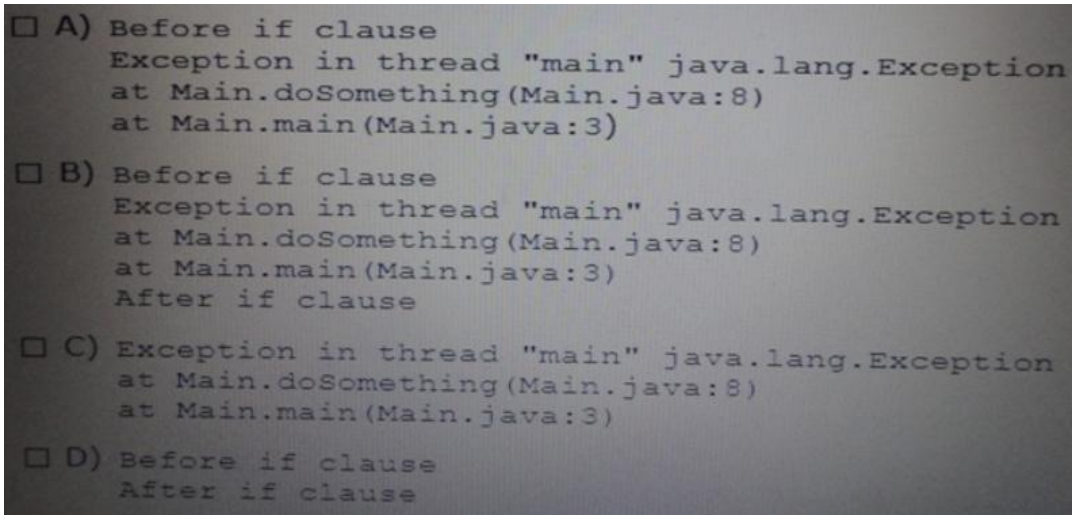
Question No : 63

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

Answer: A,D

Explanation:

The first println statement, System.out.println("Before if clause");, will always run.
If Math.Random() > 0.5 then there is an exception. The exception message is displayed and the program terminates.
If Math.Random() > 0.5 is false, then the second println statement runs as well.

Question No : 64

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

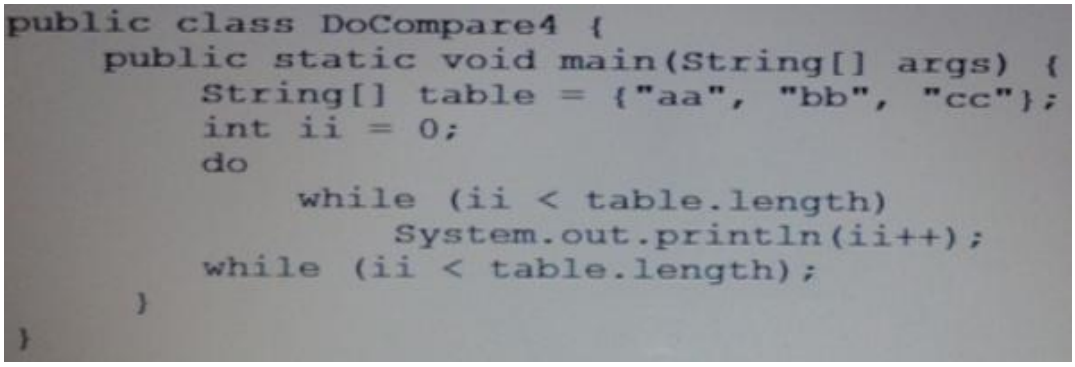
Answer: A

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 No : 65

Given:



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

What is the result?

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

Answer: B

Explanation:

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

Question No : 66

Given the code fragment:

System.out.println(2 + 4 * 9 - 3); //Line 21

System.out.println((2 + 4) * 9 - 3); // Line 22

System.out.println(2 + (4 * 9) - 3); // Line 23

System.out.println(2 + 4 * (9 - 3)); // Line 24

System.out.println((2 + 4 * 9) - 3); // Line 25

Which line of codes prints the highest number?

A. Line 21
B. Line 22
C. Line 23
D. Line 24
E. Line 25

Answer: B

Explanation: The following is printed:
35
51
35
26
35

Question No : 67

An unchecked exception occurs in a method dosomething()

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

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

Answer: D

Explanation:

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

Question No : 68

Given:

```
public class Main {
    public static void main(String[] args){
        doSomething();
    }
    private static void doSomething() {
        doSomethingElse();
    }
    private static void doSomethingElse() {
        throw new Exception();
    }
}
```

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

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

Answer: A

Explanation:

We need to catch the exception in the doSomethingElse() method.
Such as:

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

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

Question No : 69

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

Answer: B

Explanation: 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 No : 70

Which three statements are benefits of encapsulation?

A. Allowsa 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

Answer: A,B,D

Question No : 71

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

A. A class can have only one private constructor.