Oracle

Exam 1z0-804

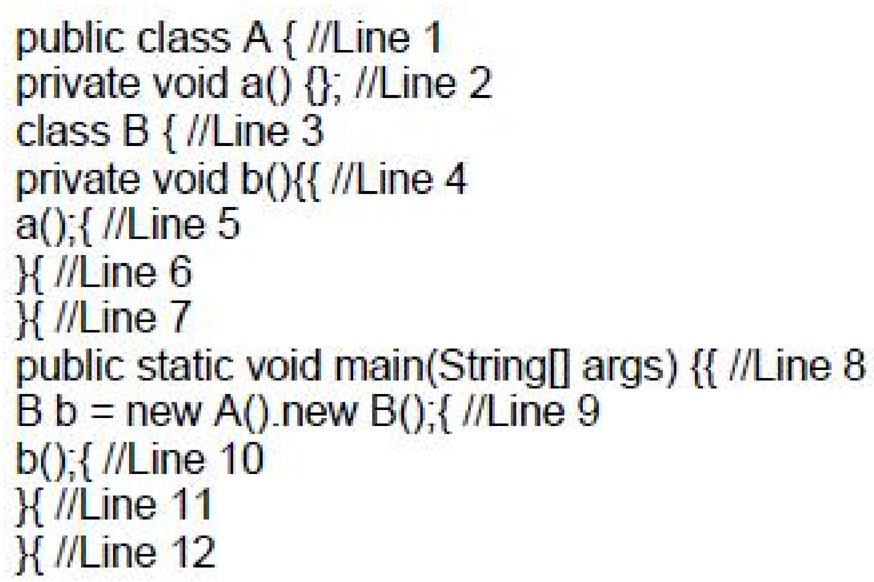
Java SE 7 Programmer II

Verson: Demo

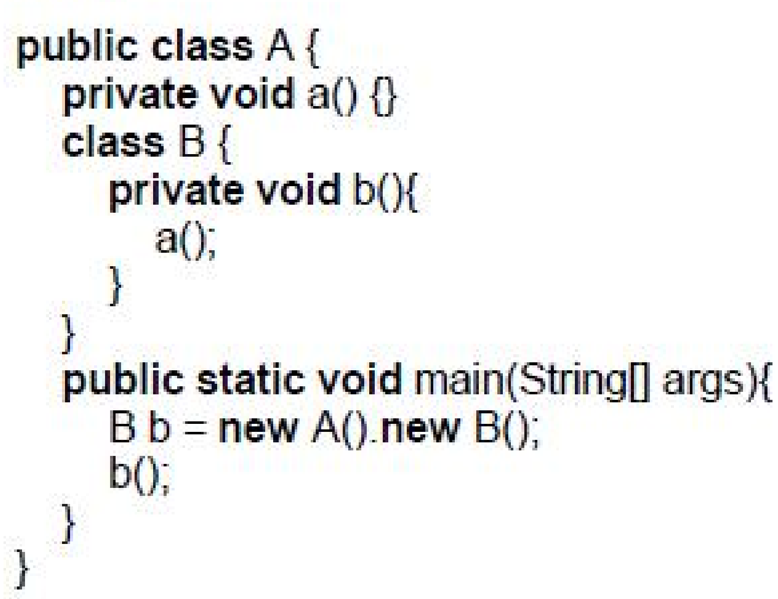
[Total Questions: 10]

Question No : 1

Given:



```
public class A { //Line 1
private void a() {}; //Line 2
class B { //Line 3
private void b(){{ //Line 4
a();{ //Line 5
}{ //Line 6
}{ //Line 7
public static void main(String[] args) {{ //Line 8
B b = new A().new B();{ //Line 9
b();{ //Line 10
}{ //Line 11
}{ //Line 12
```



```
public class A {
    private void a() {}
    class B {
        private void b(){
            a();
        }
    }
    public static void main(String[] args){
        B b = new A().new B();
        b();
    }
}
```

What is the result?

A. Compilation fails at line 9
B. Compilation fails at line 10
C. Compilation fails at line 5
D. Compilation fails at line 3
E. Compilation succeeds

Answer: B

Question No : 2

Which two codes correctly represent a standard language locale code?

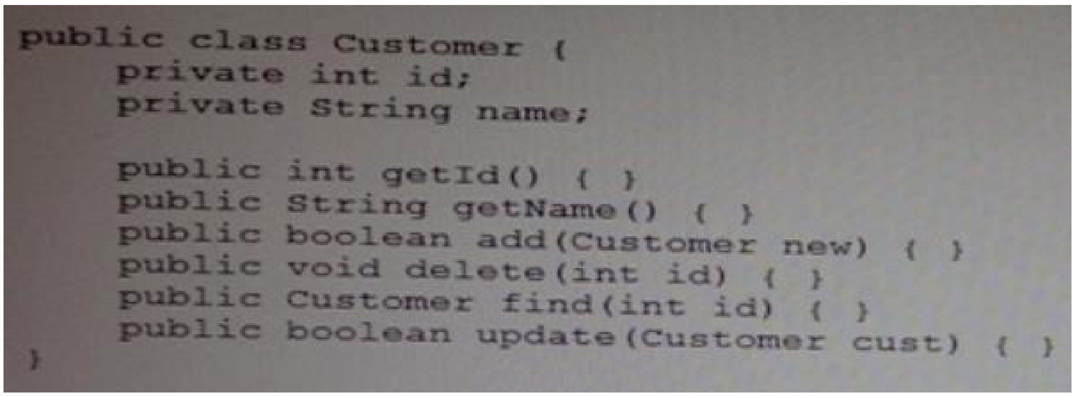
A. ES
B. FR
C. U8
D. Es
E. fr
F. u8

Answer: A,B

Explanation:
Language codes are defined by ISO 639, an international standard that assigns two- and three-letter codes tomost languages of the world. Locale uses the two-letter codes to identify the target language.

Question No : 3

Given:



```
public class Customer {
    private int id;
    private String name;

    public int getId() { }
    public String getName() { }
    public boolean add(Customer new) { }
    public void delete(int id) { }
    public Customer find(int id) { }
    public boolean update(Customer cust) { }
}
```

What two changes should you make to apply the DAO pattern to this class?

A. Make the Customer class abstract.
B. Make the customer class an interface.
C. Move the add, delete, find, and update methods into their own implementation class.
D. Create an interface that defines the signatures of the add, delete, find, and update methods.

E. Make the add, delete, and find, and update methods private for encapsulation.
F. Make the getName and getID methods private for encapsulation.

Answer: C,D

Explanation:

C:The methods related directly to the entity Customer is moved to a new class.

D: Example (here Customer is the main entity):

public class Customer {

private final String id;

private String contactName;

private String phone;

public void setId(String id) { this.id = id; }

public String getId() { return this.id; }

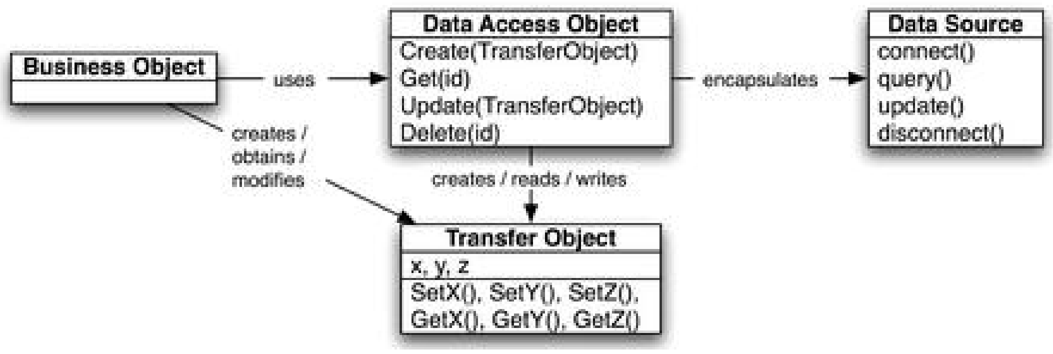
public void setContactName(String cn) { this.contactName = cn;} public String getContactName() { return
this.contactName; } public void setPhone(String phone) { this.phone = phone; } public String getPhone()
{ return this.phone; }
}

public interface CustomerDAO {

public void addCustomer(Customer c) throws DataAccessException; public Customer getCustomer(String id)throws DataAccessException; public List getCustomers() throws DataAccessException; public void
removeCustomer(String id) throws DataAccessException; public void modifyCustomer(Customer c) throws
DataAccessException; }

Note: DAO Design Pattern

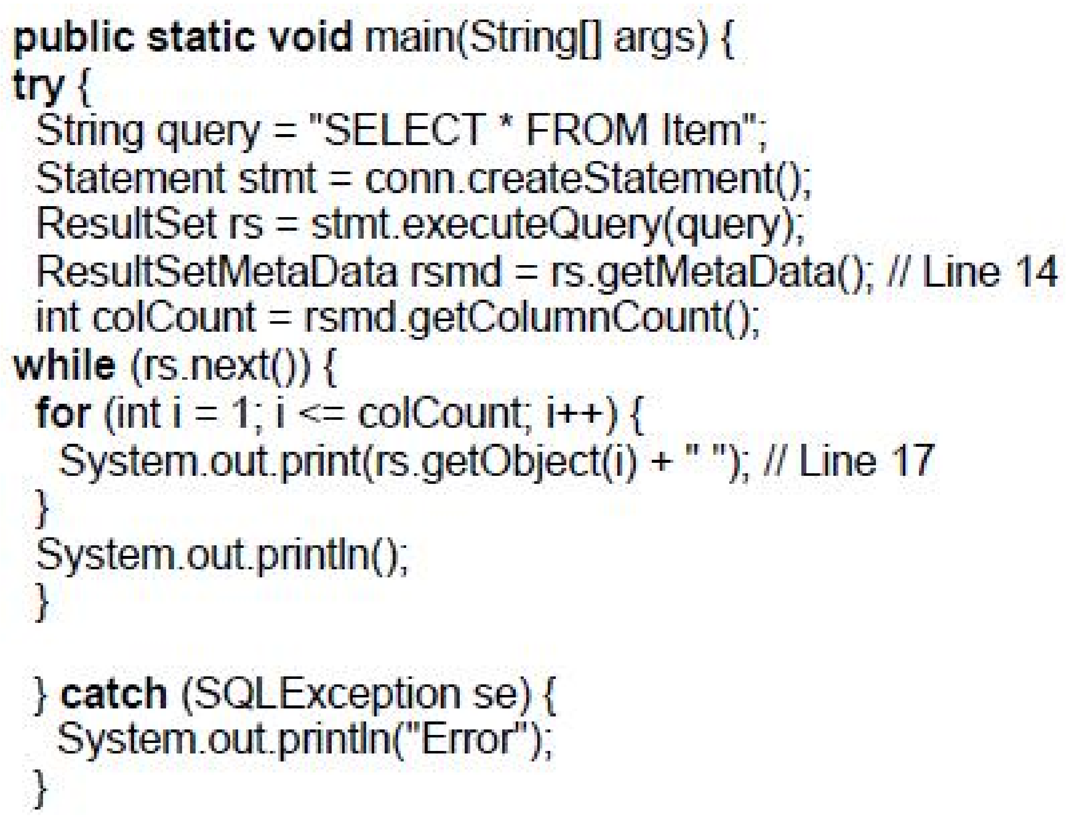
*Abstracts and encapsulates all access to a data source *Manages the connection to the data source to obtainand store data *Makes the code independent of the data sources and data vendors (e.g. plain-text, xml, LDAP,
MySQL, Oracle, DB2)



D:\Documents and Settings\useralbo\Desktop\1.jpg

Question No : 4

Given this code fragment:



```
public static void main(String[] args) {
try {
  String query = "SELECT * FROM Item";
  Statement stmt = conn.createStatement();
  ResultSet rs = stmt.executeQuery(query);
  ResultSetMetaData rsmd = rs.getMetaData(); // Line 14
  int colCount = rsmd.getColumnCount();
while (rs.next()) {
  for (int i = 1; i <= colCount; i++) {
    System.out.print(rs.getObject(i) + " "); // Line 17
  }
  System.out.println();
  }

  } catch (SQLException se) {
    System.out.println("Error");
  }
```

Assume that the SQL query returns records.

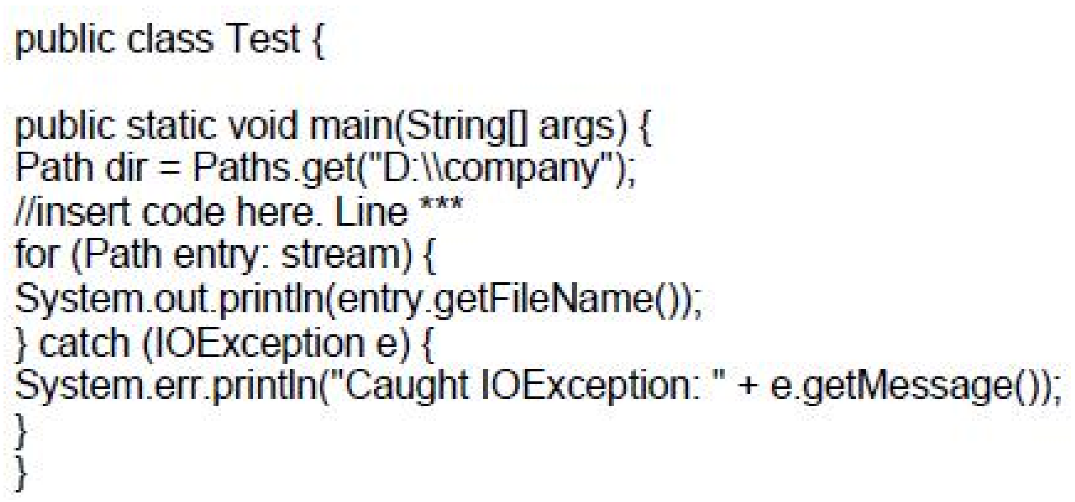
What is the result?

A. Compilation fails due to error at line 17
B. The program prints Error
C. The program prints each record
D. Compilation fails at line 14

Answer: C

Question No : 5

Given the code fragment:



```
public class Test {

public static void main(String[] args) {
Path dir = Paths.get("D:\\company");
//insert code here. Line ***
for (Path entry: stream) {
System.out.println(entry.getFileName());
} catch (IOException e) {
System.err.println("Caught IOException: " + e.getMessage());
}
}
```

Which two try statements, when inserted at line ***, enable you to print files with the extensions.java, .htm, and.jar.

A. try (DirectoryStream<Path> stream = Files.newDirectoryStream(dir,"*.{java,htm,jar}")){
B. try (DirectoryStream<Path> stream = Files.newDirectoryStream(dir,"*. [java,htm,jar]")) {
C. try (DirectoryStream<Path> stream = Files.newDirectoryStream(dir,"*.{java*,htm*,jar*}"))
{
D. try (DirectoryStream<Path> stream = Files.newDirectoryStream(dir,"**.{java,htm,jar}")) {

Answer: A,D

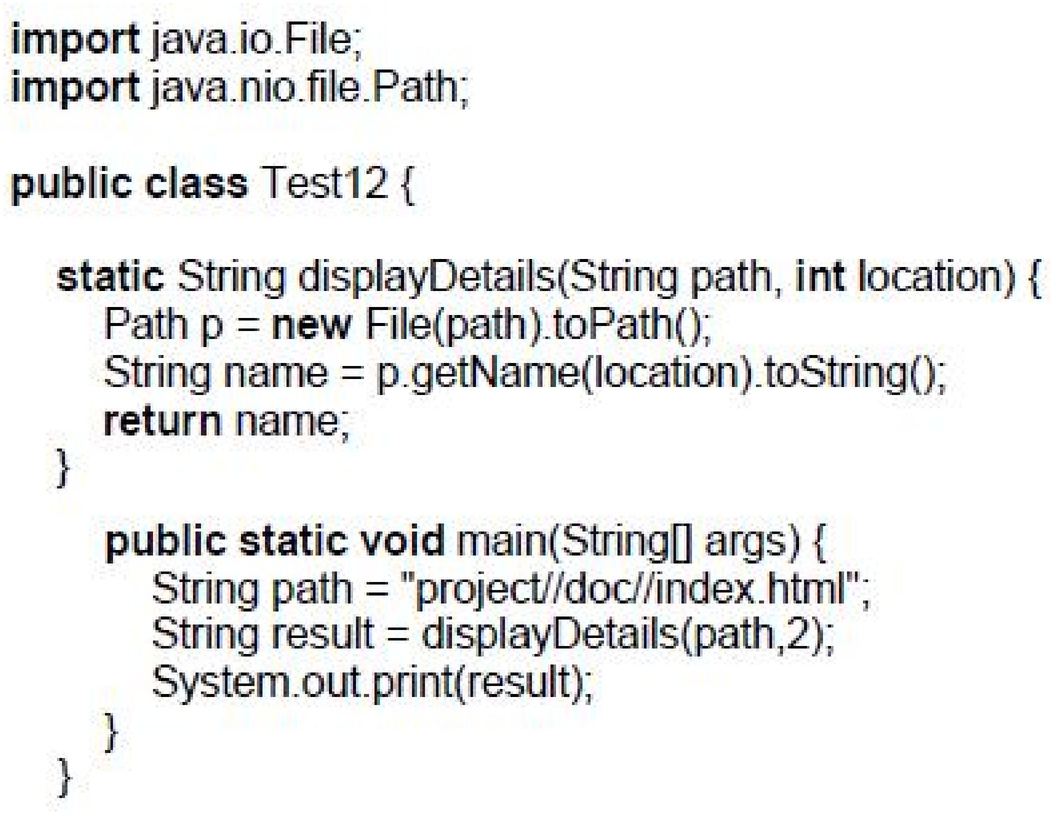
Explanation:

"*. {java,htm,jar} and

"**. {java,htm,jar} will match any file with file endings java, htm, or jar.

Question No : 6

Given:



```
import java.io.File;
import java.nio.file.Path;

public class Test12 {

  static String displayDetails(String path, int location) {
    Path p = new File(path).toPath();
    String name = p.getName(location).toString();
    return name;
  }

    public static void main(String[] args) {
      String path = "project//doc//index.html";
      String result = displayDetails(path,2);
      System.out.print(result);
    }
  }
```

What is the result?

A. doc
B. index.html
C. an IllegalArgumentException is thrown at runtime.
D. An InvalidPthException is thrown at runtime.
E. Compilation fails.

Answer: B

Explanation:

p.getName(int location) = returns path' name element by index/location (starts with 0)

Example:

path = "project//doc//index.html"

p.getName(0) = project

p.getName(1) = doc

p.getName(2) = index.html

Question No : 7

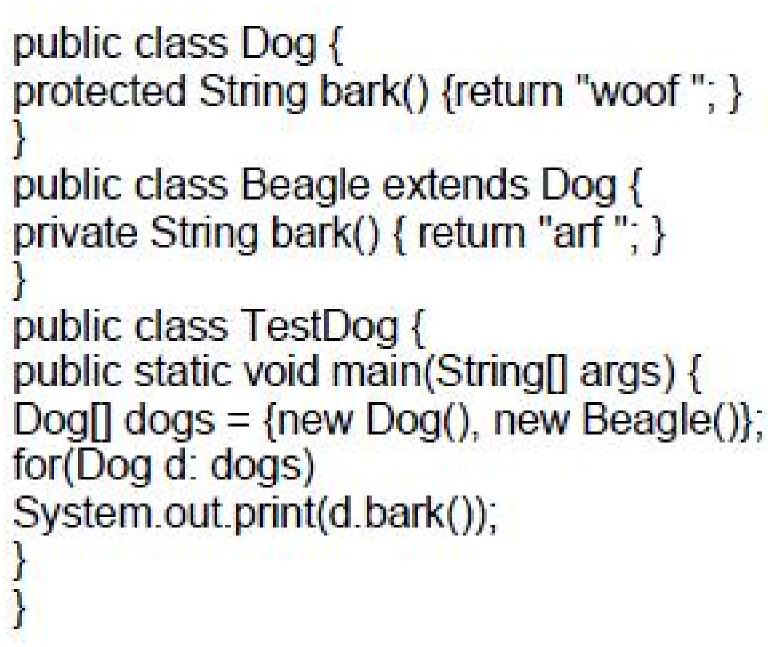
Which two compile?

A. interface Compilable {
void compile();
}
B. interface Compilable {
final void compile();
}
C. interface Compilable {
static void compile();
}
D. interface Compilable {
abstract void compile();
}
E. interface Compilable {
protected abstract void compile ();
}

Answer: A,D

Question No : 8

Given:



```
public class Dog {
protected String bark() {return "woof "; }
}
public class Beagle extends Dog {
private String bark() { return "arf "; }
}
public class TestDog {
public static void main(String[] args) {
Dog[] dogs = {new Dog(), new Beagle()};
for(Dog d: dogs)
System.out.print(d.bark());
}
}
```

What is the result?

A. woof arf
B. woof woof
C. arf arf
D. A RuntimeException is generated
E. The code fails to compile

Answer: E

Explanation:

class Dog {

protected String bark()

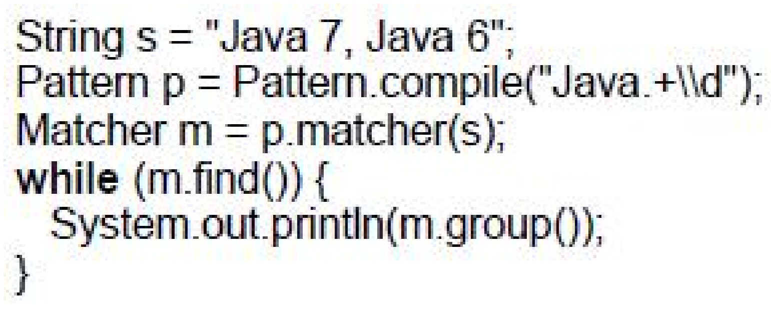
public class Beagle extends Dog {

private String bark()

Cannot reduce the visibility of the inherited method from Dog

Question No : 9

Given the code fragment:



```
String s = "Java 7, Java 6";
Pattern p = Pattern.compile("Java.+\\d");
Matcher m = p.matcher(s);
while (m.find()) {
   System.out.println(m.group());
}
```

What is the result?

A. Java 7
B. Java 6
C. Java 7, Java 6
D. Java 7
java 6
E. Java

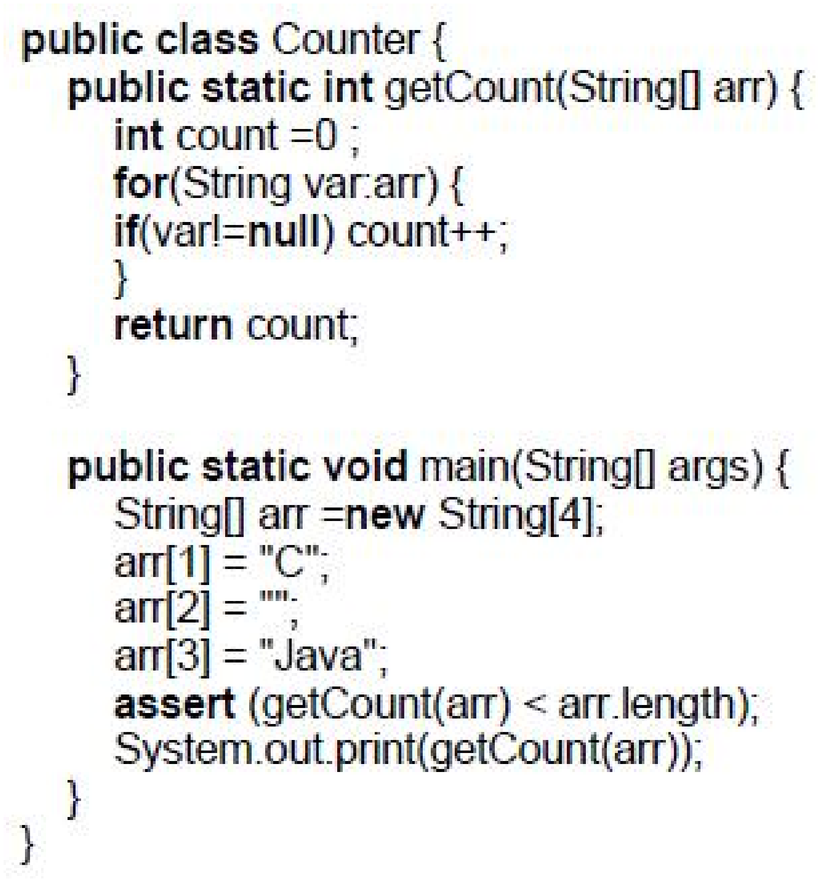
Answer: C

Explanation:

regex: Java / one or more anything !!! / ends with a digit so it is the source string

Question No : 10

Given:



```
public class Counter {
    public static int getCount(String[] arr) {
        int count =0 ;
        for(String var:arr) {
        if(var!=null) count++;
        }
        return count;
    }

    public static void main(String[] args) {
        String[] arr =new String[4];
        arr[1] = "C";
        arr[2] = "";
        arr[3] = "Java";
        assert (getCount(arr) < arr.length);
        System.out.print(getCount(arr));
    }
}
```

And the commands:

javac Counter.java

java ea Counter

What is the result?

A. 2
B. 3
C. NullPointException is thrown at runtime

D. AssertionError is thrown at runtime
E. Compilation fails

Answer: B

Explanation:

The command line javac Counter.java

Willcompile the code.

The command line java ea Counter

Willrun the cod with assertions enabled.

Assertion is true because getCount(arr) = 3 and Length of array is 4

The following line:

assert (getCount(arr) < arr.length);

where the Boolean expression getCount(arr) < arr.length will evaluate to false, will ensure that anAssertionError is thrown at runtime.

Note:The javac command compiles Java source code into Java bytecodes. You then use the Java interpreter -the java command - to interprete the Java bytecodes.

Note 2:The java tool launches a Java application. It does this by starting a Java runtime environment, loading aspecified class, and invoking that class's main method. The method declaration must look like the following:public static void main(String args[])

Paramater ea:

-enableassertions[:<package name>"..." | :<class name>] -ea[:<package name>"..." | :<class name>]

Enable assertions. Assertions are disabled by default. With no arguments, enableassertions or -ea enablesassertions.

Note 3:

An assertion is a statement in the JavaTM programming language that enables you to test your assumptionsabout your program.

Each assertion contains a boolean expression that you believe will be true when the assertion executes. If it isnot true, the system will throw an error.