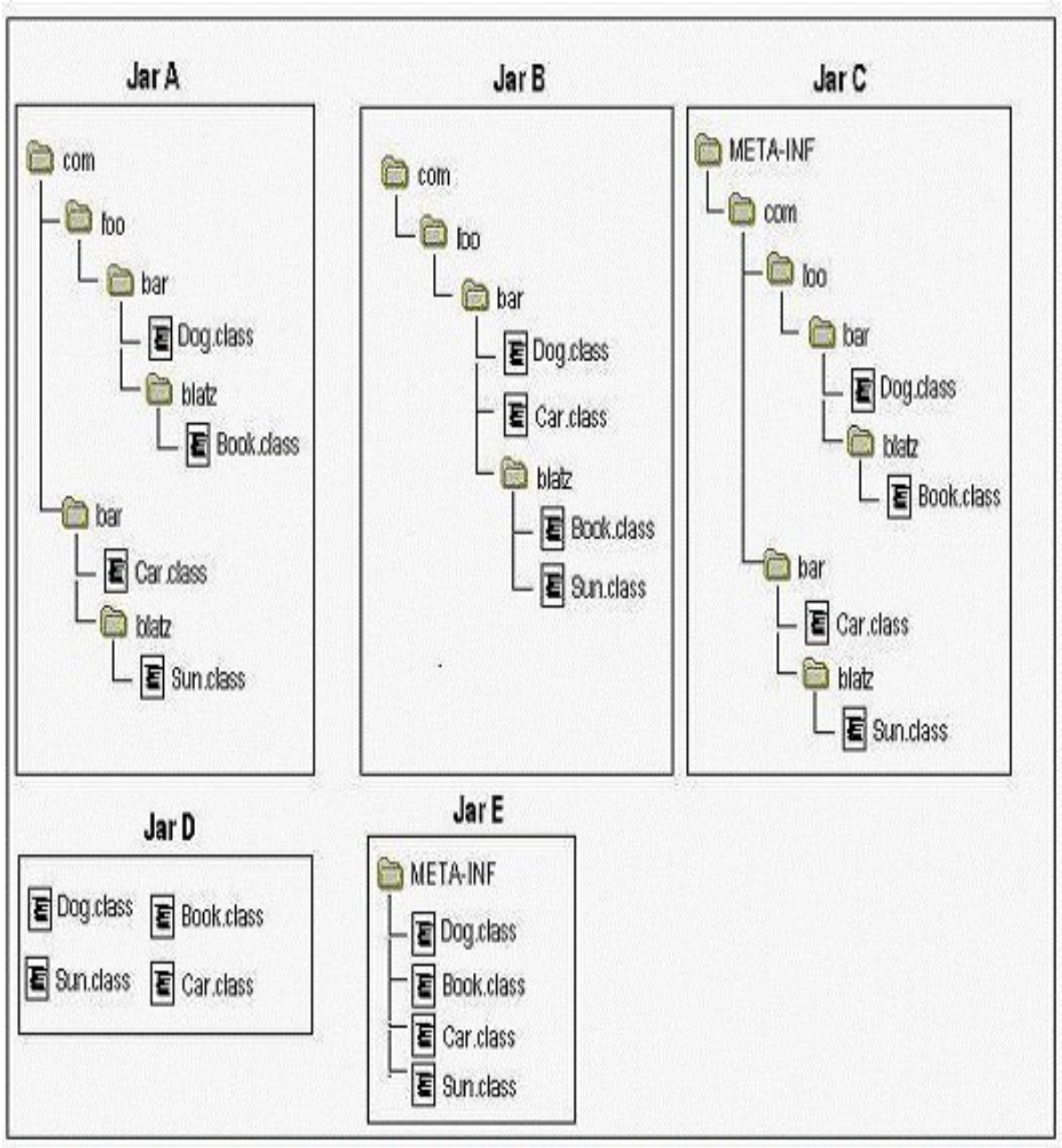
Development

For more details on SUN Certifications, visit JavaScjpDumps

Q: 01 Click the Exhibit button.
Given the fully-qualified class names:
com.foo.bar.Dog
com.foo.bar.blatz.Book
com.bar.Car
com.bar.blatz.Sun
Which graph represents the correct directory structure for a JAR file from which those classes can be used by the compiler and JVM?



A. Jar A
B. Jar B
C. Jar C
D. Jar D
E. Jar E
Answer: A

Q: 02
A class games.cards.Poker is correctly defined in the jar file Poker.jar. A user wants to execute the main method of Poker on a UNIX system using the command:

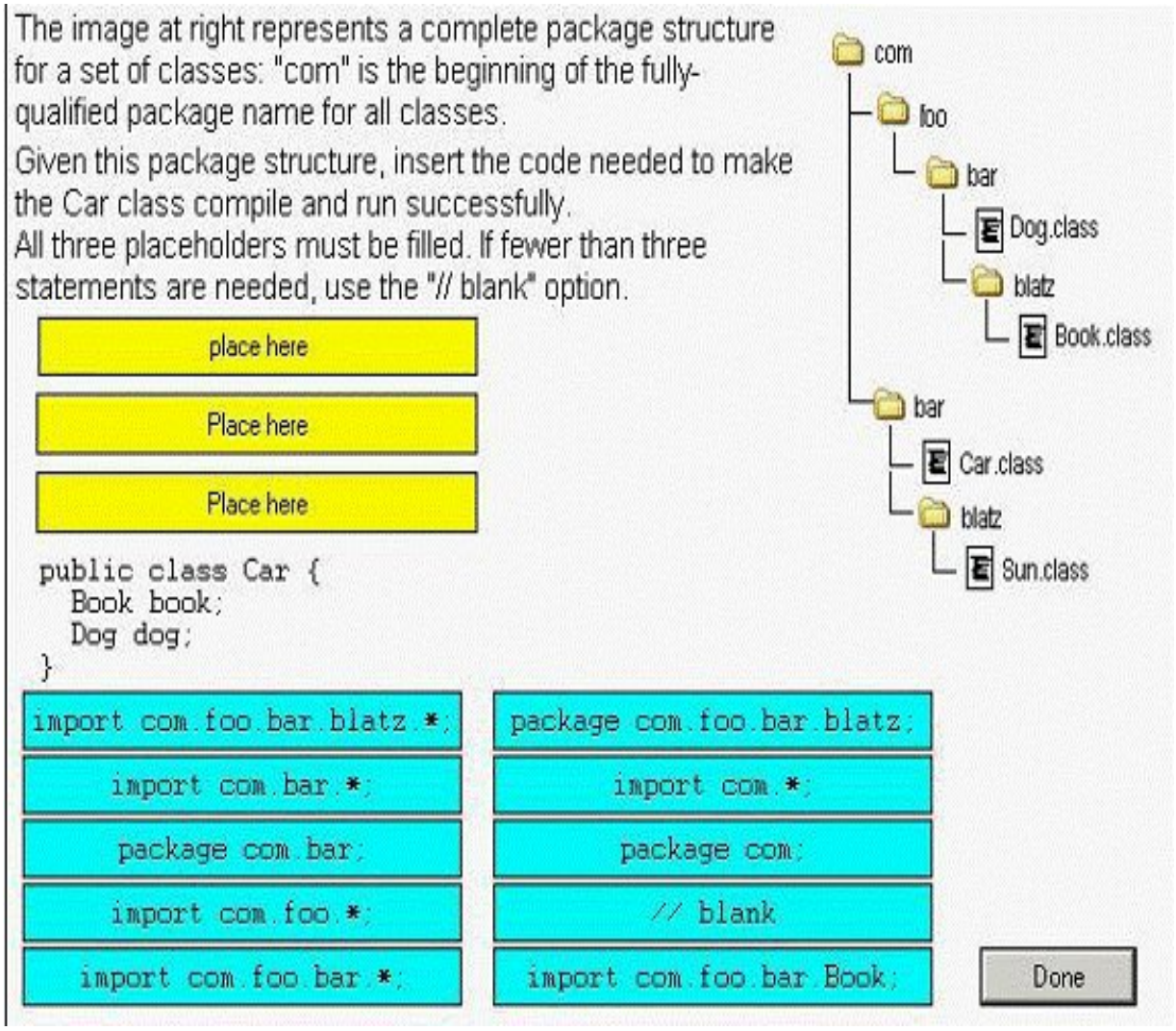
java games.cards.PokerWhat allows the user to do this?
A. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java
B. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java/*.jar
C. Put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java/Poker.jar
D. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java
E. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java/*.jar
F. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java/Poker.jar
Answer: C

Q: 03 Given:
11. public class Commander {
12. public static void main(String[] args) {
13. String myProp = /* insert code here */
14. System.out.println(myProp);
15. }
16. }
and the command line:
java -Dprop.custom=gobstopper Commander
Which two, placed on line 13, will produce the output gobstopper? (Choose two.)
A. System.load("prop.custom");
B. System.getenv("prop.custom");
C. System.property("prop.custom");
D. System.getProperty("prop.custom");
E. System.getProperties().getProperty("prop.custom");
Answer: D, E

Q: 04
A developer is creating a class Book, that needs to access class Paper. The Paper class is deployed in a JAR named myLib.jar. Which three, taken independently, will allow thedeveloper to use the Paper class while compiling the Book class? (Choose three.)
A. The JAR file is located at $JAVA_HOME/jre/classes/myLib.jar.
B. The JAR file is located at $JAVA_HOME/jre/lib/ext/myLib.jar..
C. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes
/foo/myLib.jar/Paper.class.
D. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes
/foo/myLib.jar.
E. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -cp /foo/myLib.jar/Paper Book.java.

F. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -d /foo/myLib.jar
Book.java
G. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -classpath
/foo/myLib.jar Book.java
Answer: B, D, G

Q: 05 Click the Task button.



Solution:

1. **package com.bar;**
2. **import com.foo.bar.*;**
3. **import com.foo.bar.blatz.*;**

Q: 06 Given:
1. package com.company.application;
2.

3. public class MainClass {
4. public static void main(String[] args) {}
5. }
And MainClass exists in the /apps/com/company/application directory. Assume the CLASSPATH
environment variable is set to "." (current directory).
Which two java commands entered at the command line will run MainClass? (Choose two.)
A. java MainClass if run from the /apps directory
B. java com.company.application.MainClass if run from the /apps directory
C. java -classpath /apps com.company.application.MainClass if run from any directory
D. java -classpath . MainClass if run from the /apps/com/company/application directory
E. java -classpath /apps/com/company/application:. MainClass if run from the /apps directory
F. java com.company.application.MainClass if run from the /apps/com/company/application directory
Answer: B, C

Q: 07 Given a correctly compiled class whose source code is:
1. package com.sun.sjcp;
2. public class Commander {
3. public static void main(String[] args) {
4. // more code here
5. }
6. }
Assume that the class file is located in /foo/com/sun/sjcp/, the current directory is /foo/, and that the classpath contains "." (current directory).
Which command line correctly runs Commander?
A. java Commander
B. java com.sun.sjcp.Commander
C. java com/sun/sjcp/Commander
D. java -cp com.sun.sjcp Commander
E. java -cp com/sun/sjcp Commander
Answer: B

Q: 08
A UNIX user named Bob wants to replace his chess program with a new one, but he is not sure where the old one is installed. Bob is currently able to run a Java chess program starting from his home directory /home/bob using the command:

java -classpath /test:/home/bob/downloads/*.jar games.
Chess Bob's CLASSPATH is set (at login time) to:
/usr/lib:/home/bob/classes:/opt/java/lib:/opt/java/lib/*.jar
What is a possible location for the Chess.class file?
A. /test/Chess.class

B. /home/bob/Chess.class
C. /test/games/Chess.class
D. /usr/lib/games/Chess.class
E. /home/bob/games/Chess.class
F. inside jarfile /opt/java/lib/Games.jar (with a correct manifest)
G. inside jarfile /home/bob/downloads/Games.jar (with a correct manifest)
Answer: C

09. Given these classes in different files:

```
package xcom;
public class Useful {
int increment(int x) { return ++x; }
}
import xcom.*; // line 1
class Needy3 {
public static void main(String[] args) {
xcom.Useful u = new xcom.Useful(); // line 2
System.out.println(u.increment(5));
}
}
```

Which statements are true? (Choose all that apply.)

A. The output is 0.
B. The output is 5.
C. The output is 6.
D. Compilation fails.
E. The code compiles if line 1 is removed.
F. The code compiles if line 2 is changed to read
Useful u = new Useful();
Answer:
-> **D** is correct. The increment() method must be marked public to be accessed outside of the package. If increment() was public, **C, E,** and **F** would be correct..
-> **A** and **B** are incorrect output, even if increment() is public.

10. Given the following directory structure:

```
org
| -- Robot.class
|
| -- ex
        |-- Pet.class
        |
        |-- why
        |-- Dog.class
```

And the following source file:

```
class MyClass {
Robot r;
```

Pet p;
Dog d;
}
Which statement(s) *must* be added for the source file to compile? (Choose all that apply.)
A. package org;
B. import org.*;
C. package org.*;
D. package org.ex;
E. import org.ex.*;
F. package org.ex.why;
G. package org.ex.why.Dog;
Answer:
-> **B , E,** and **F** are required. The only way to access class Dog is via **F**, which is a package statement. Since you can have only one package statement in a source file, you have to get access to class Robot and class Pet using import statements. Option **B** accesses Robot, and option **E** accesses Pet.
-> **A, C, D,** and **G** are incorrect based on the above. Also, **C** and **G** are incorrect syntax.

11. Given:
1. // insert code here
2. class StatTest {
3. public static void main(String[] args) {
4. System.out.println(Integer.MAX_VALUE);
5. }
6. }
Which, inserted independently at line 1, compiles? (Choose all that apply.)
A. import static java.lang;
B. import static java.lang.Integer;
C. import static java.lang.Integer.*;
D. import static java.lang.Integer.*_VALUE;
E. import static java.lang.Integer.MAX_VALUE;
F. None of the above statements are valid import syntax.
Answer:
->**C** and **E** are correct syntax for static imports. Line 4 isn't making use of static imports, so the code will also compile with none of the imports.
->**A, B, D,** and **F** are incorrect based on the above.

12. Given the default classpath:
/foo
And this directory structure:
foo
|
test
|

xcom
|--A.class
|--B.java

And these two files:

```
package xcom;
public class A { }
```

```
package xcom;
public class B extends A { }
```

Which allows B.java to compile? (Choose all that apply.)

A. Set the current directory to xcom then invoke
javac B.java
B. Set the current directory to xcom then invoke
javac -classpath . B.java
C. Set the current directory to test then invoke
javac -classpath . xcom/B.java
D. Set the current directory to test then invoke
javac -classpath xcom B.java
E. Set the current directory to test then invoke
javac -classpath xcom:. B.java

Answer:

-> **C** is correct. In order for B.java to compile, the compiler first needs to be able to find B.java. Once it's found B.java it needs to find A.class. Because A.class is in the xcom package the compiler won't find A.class if it's invoked from the xcom directory. Remember that the -classpath isn't looking for B.java, it's looking for whatever classes B.java needs (in this case A.class).
->**A, B,** and **D** are incorrect based on the above. **E** is incorrect because the compiler can't find B.java.

13. Given two files:

```
package xcom;
public class Stuff {
public static final int MY_CONSTANT = 5;
public static int doStuff(int x) { return (x++)*x; }
}
```

```
import xcom.Stuff.*;
import java.lang.System.out;
class User {
public static void main(String[] args) {
new User().go();
}
void go() { out.println(doStuff(MY_CONSTANT)); }
}
```

What is the result?

A. 25
B. 30

C. 36
D. Compilation fails.
E. An exception is thrown at runtime.
Answer:
-> **D** is correct. To import static members, an import statement must begin: import static.
->**A, B, C,** and **E** are incorrect based on the above.

14. Given three files:

```
package xcom;
public class A {
// insert code here
}
```

```
package xcom;
public class B extends A {public void doB() { System.out.println("B.doB"); } }
```

```
import xcom.B;
class TestXcom {
public static void main(String[] args) {
B b = new B(); b.doB(); b.go();
}
}
```

Which, inserted at // insert code here will allow all three files to compile? (Choose all that apply.)
A. void go() { System.out.println("a.go"); }
B. public void go() { System.out.println("a.go"); }
C. private void go() { System.out.println("a.go"); }
D. protected void go() { System.out.println("a.go"); }
E. None of these options will allow the code to compile.
Answer:
-> **B** is correct. The public access modifier is the only one that allows code from outside a package to access methods in a package—regardless of inheritance.
->**A, B, D,** and **E** are incorrect based on the above.

15. Given:

```
class TestProps {
public static void main(String[] args) {
String s = System.getProperty("aaa","bbb");
}
}
```

And the command-line invocation:
java -Daaa=ccc TestProps
What is always true? (Choose all that apply.)
A. The value of property aaa is aaa.
B. The value of property aaa is bbb.
C. The value of property aaa is ccc.

D. The value of property bbb is aaa.
E. The value of property bbb is ccc.
F. The invocation will not complete without error.
Answer:
-> **C** is correct. The value of aaa is set at the command line. If aaa had no value when getProperty was invoked, then aaa would have been set to bbb.
->**A, B, D, E,** and **F** are incorrect based on the above.

16. If three versions of MyClass.java exist on a file system:
Version 1 is in /foo/bar
Version 2 is in /foo/bar/baz
Version 3 is in /foo/bar/baz/bing
And the system's classpath includes:
/foo/bar/baz
And this command line is invoked from /foo
javac -classpath /foo/bar/baz/bing:/foo/bar MyClass.java
Which version will be used by javac?
A. /foo/MyClass.java
B. /foo/bar/MyClass.java
C. /foo/bar/baz/MyClass.java
D. /foo/bar/baz/bing/MyClass.java
E. The result is not predictable.
Answer:
->**D** is correct. A -classpath included with a javac invocation overrides a system classpath. When javac is using any classpath, it reads the classpath from left to right, and uses the first match it finds.
->**A, B, C,** and **E** are incorrect based on the above.

17. Which are true? (Choose all that apply.)
A. The java command can access classes from more than one package, from a single JAR file.
B. JAR files can be used with the java command but not with the javac command.
C. In order for JAR files to be used by java, they MUST be placed in the /jre/lib/ext subdirectory within the J2SE directory tree.
D. In order to specify the use of a JAR file on the command line, the JAR file's path and filename MUST be included.
E. When a part of a directory tree that includes subdirectories with files is put into a JAR file, all of the files are saved in the JAR file, but the subdirectory structure is lost.

Answer:
-> **A** and **D** are correct.
->**B** is incorrect because javac can also use JAR files. **C** is incorrect because JARs can be located in ../jre/lib/ext, but they can also be accessed if they live in other locations.
E is incorrect, JAR files maintain directory structures.

18. Given two files:

```
package pkg;
public class Kit {
public String gluelt(String a, String b) { return a+b; }
}
import pkg.*;
class UseKit {
public static void main(String[] args) {
String s = new Kit().gluelt(args[1], args[2]);
System.out.println(s);
}
}
```

And the following sub-directory structure:

```
test
        |--UseKit.class
        |
        com
        |--KitJar.jar
```

If the current directory is test, and the file pkg/Kit.class is in KitJar.jar, which command line will produce the output bc ? (Choose all that apply.)

A. java UseKit b c
B. java UseKit a b c
C. java -classpath com UseKit b c
D. java -classpath com:. UseKit b c
E. java -classpath com/KitJar.jar UseKit b c
F. java -classpath com/KitJar.jar UseKit a b c
G. java -classpath com/KitJar.jar:. UseKit b c
H. java -classpath com/KitJar.jar:. UseKit a b c

Answer:

->**H** is correct.

->**A, C, E,** and **G** are incorrect if for no other reason than args[] is 0-based.

B, D, and **F** are incorrect because java needs a classpath that specifies two directories, one for the class file (the . directory), and one for the JAR file (the com directory). Remember, to find a JAR file, the c