Section A

----------------

QUESTION 1

Given:

2. public class Threads2 implements Runnable {

3.

4.

public void run() {

5.

System.out.println("run.");

6.

throw new RuntimeException("Problem");

7.

}

8.public static void main(String[] args) {

9.Thread t = new Thread(new Threads2());

10.t.start();

11.System.out.println("End of method.");

12.}

13. }

Which two can be results? (Choose two.)

A. java.lang.RuntimeException: Problem

B. run.

java.lang.RuntimeException: Problem

C. End of method.

java.lang.RuntimeException: Problem

D. End of method.

run.

java.lang.RuntimeException: Problem

E. run.

java.lang.RuntimeException: Problem

End of method.

QUESTION 2

Which two statements are true? (Choose two.)

A. It is possible for more than two threads to deadlock at once.

B. The JVM implementation guarantees that multiple threads cannot enter into a deadlocked state.

C. Deadlocked threads release once their sleep() method's sleep duration has expired.

D. Deadlocking can occur only when the wait(), notify(), and notifyAll() methods are used incorrectly.

E. It is possible for a single-threaded application to deadlock if synchronized blocks are used incorrectly.

F.If a piece of code is capable of deadlocking, you cannot eliminate the possibility of deadlocking by inserting invocations of Thread.yield().

QUESTION 3

Given:

void waitForSignal() {

Object obj = new Object();

synchronized (Thread.currentThread()) {

obj.wait();

obj.notify();

}

}

Which statement is true?

1. This code can throw an InterruptedException.
2. This code can throw an IllegalMonitorStateException.
3. This code can throw a TimeoutException after ten minutes.
4. Reversing the order of obj.wait() and obj.notify() might cause this method to complete normally.
5. A call to notify() or notifyAll() from another thread might cause this method to complete normally.
6. This code does NOT compile unless "obj.wait()" is replaced with "((Thread) obj).wait()".

QUESTION 4

Click the Exhibit button.

What is the output if the main() method is run?

1. public class Starter extends Thread {

2.

private int x = 2;

3.

public static void main(String[] args) throws Exception {

4.

new Starter().makeItSo();

5.}

6.public Starter(){

7.x = 5;

8.start();

9.}

10.public void makeItSo() throws Exception {

11.join();

12. x = x - 1;

13. System.out.println(x);

14. }

15. public void run() { x \*= 2; }

16. }

1. 4
2. 5
3. 8
4. 9
5. Compilation fails.
6. An exception is thrown at runtime.
7. It is impossible to determine for certain.

QUESTION 5

Given:

1. class PingPong2 {

2. synchronized void hit(long n) {

3. for(int i = 1; i < 3; i++)

4. System.out.print(n + "-" + i + " ");

5. }

6. }

1. public class Tester implements Runnable {

2. static PingPong2 pp2 = new PingPong2();

3. public static void main(String[] args) {

4. new Thread(new Tester()).start();

5. new Thread(new Tester()).start();

6. }

7. public void run() { pp2.hit(Thread.currentThread().getId()); }

8. }

Which statement is true?

1. The output could be 5-1 6-1 6-2 5-2
2. The output could be 6-1 6-2 5-1 5-2
3. The output could be 6-1 5-2 6-2 5-1
4. The output could be 6-1 6-2 5-1 7-1

QUESTION 6

Given:

1. public class Threads4 {

2. public static void main (String[] args) {

3. new Threads4().go();

4. }

5. public void go() {

6. Runnable r = new Runnable() {

7. public void run() {

8. System.out.print("foo");

9. }

10. };

11. Thread t = new Thread(r);

12. t.start();

13. t.start();

14. }

15. }

What is the result?

1. Compilation fails.
2. An exception is thrown at runtime.
3. The code executes normally and prints "foo".
4. The code executes normally, but nothing is printed.

QUESTION 7

Given:

1. public abstract class Shape {

2. private int x;

3. private int y;

4. public abstract void draw();

5. public void setAnchor(int x, int y) {

6. this.x = x;

7. this.y = y;

8. }

9. }

Which two classes use the Shape class correctly? (Choose two.)

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 \*/}

}

QUESTION 8

Given:

1. public class Barn {

2. public static void main(String[] args) {

3. new Barn().go("hi", 1);

4. new Barn().go("hi", "world", 2);

5. }

6. public void go(String... y, int x) {

7. System.out.print(y[y.length - 1] + " ");

8. }

9. }

What is the result?

1. hi hi
2. hi world
3. world world
4. Compilation fails.
5. An exception is thrown at runtime.

QUESTION 9

Given:

1. class Nav{

2.

public enum Direction { NORTH, SOUTH, EAST, WEST }

3. }

1. public class Sprite{

2.

//

insert code here

3. }

Which code, inserted at line 14, allows the Sprite class to compile?

1. Direction d = NORTH;
2. Nav.Direction d = NORTH;
3. Direction d = Direction.NORTH;
4. Nav.Direction d = Nav.Direction.NORTH;

QUESTION 10

1. public interface A {

2. public void doSomething(String thing);

3. }

1. public class AImpl implements A {

2. public void doSomething(String msg) {}

3. }

1. public class B {

2. public A doit(){

3. //more code here

4. }

5. public String execute(){

6. //more code here

7. }

8. }

1. public class C extends B {

2.

public AImpl doit(){

3. //more code here

4. }

5.

6. public Object execute() {

7. //more code here

8. }

9. }

Compilation will succeed for all classes and interfaces.

Compilation of class C will fail because of an error in line 2.

Compilation of class C will fail because of an error in line 6.

Compilation of class AImpl will fail because of an error in line 2.

QUESTION 11

What is the result?

public class Person {

String name = "No name";

public Person(String nm) { name = nm; }

}

public class Employee extends Person {

String empID = "0000";

public Employee(String id) { empID = id; }

}

public class EmployeeTest {

public static void main(String[] args){

Employee e = new Employee("4321");

System.out.println(e.empID);

}

}

A. 4321

B. 0000

C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 18.

QUESTION 12

Given:

1. public class Rainbow {

2. public enum MyColor {

3. RED(0xff0000), GREEN(0x00ff00), BLUE(0x0000ff);

4. private final int rgb;

5. MyColor(int rgb) { this.rgb = rgb; }

6. public int getRGB() { return rgb; }

7. };

8. public static void main(String[] args) {

9. //insert code here

10. }

11. }

Which code fragment, inserted at line 19, allows the Rainbow class to compile?

1. MyColor skyColor = BLUE;
2. MyColor treeColor = MyColor.GREEN;
3. if(RED.getRGB() < BLUE.getRGB()) { }
4. Compilation fails due to other error(s) in the code.
5. MyColor purple = new MyColor(0xff00ff);
6. MyColor purple = MyColor.BLUE + MyColor.RED;

QUESTION 13

Given:

1. class Atom {

2. Atom() { System.out.print("atom "); }

3. }

4. class Rock extends Atom {

5. Rock(String type) { System.out.print(type); }

6. }

7. public class Mountain extends Rock {

8. Mountain() {

9. super("granite ");

10. new Rock("granite ");

11. }

12. public static void main(String[] a) { new Mountain(); }

13. }

What is the result?

1. Compilation fails.
2. atom granite
3. granite granite
4. atom granite granite
5. An exception is thrown at runtime.
6. atom granite atom granite

QUESTION 15

Given:

interface TestA { String toString(); }

public class Test {

public static void main(String[] args) {

System.out.println(new TestA() {

public String toString() { return "test"; }

});

}

}

What is the result?A.

1. test
2. null
3. An exception is thrown at runtime.
4. Compilation fails because of an error in line 1.
5. Compilation fails because of an error in line 4.
6. Compilation fails because of an error in line 5.

QUESTION 16

Given:

1. public static void parse(String str) {

2. try {

3. float f = Float.parseFloat(str);

4. } catch (NumberFormatException nfe) {

5. f = 0;

6. } finally {

7. System.out.println(f);

8. }

9. }

10. public static void main(String[] args) {

11. parse("invalid");

12. }

What is the result?

1. 0.0
2. Compilation fails.
3. A ParseException is thrown by the parse method at runtime.
4. A NumberFormatException is thrown by the parse method at runtime.

QUESTION 17

Given:

public class Blip {

protected int blipvert(int x) { return 0; }

}

class Vert extends Blip {

//

insert code here

}

Which five methods, inserted independently at line 5, will compile? (Choose five.)

1. public int blipvert(int x) { return 0; }
2. private int blipvert(int x) { return 0; }
3. private int blipvert(long x) { return 0; }
4. protected long blipvert(int x) { return 0; }
5. protected int blipvert(long x) { return 0; }
6. protected long blipvert(long x) { return 0; }
7. protected long blipvert(int x, int y) { return 0; }

QUESTION 18

Given:

1. class Super {

2. private int a;

3. protected Super(int a) { this.a = a; }

4. }

11. class Sub extends Super {

12. public Sub(int a) { super(a); }

13. public Sub() { this.a = 5; }

14. }

Which two, independently, will allow Sub to compile? (Choose two.)

A. Change line 2 to:

public int a;

B. Change line 2 to:

protected int a;

C. Change line 13 to:

public Sub() { this(5); }

D. Change line 13 to:

public Sub() { super(5); }

E. Change line 13 to:

public Sub() { super(a); }

QUESTION 19

Given:

1. package test;

2.

3. class Target {

4.

public String name = "hello";

5. }

What can directly access and change the value of the variable name?

1. any class
2. only the Target class
3. any class in the test package
4. any class that extends Target

QUESTION 21

Given:

11. abstract class Vehicle { public int speed() { return 0; }

12. class Car extends Vehicle { public int speed() { return 60; }

13. class RaceCar extends Car { public int speed() { return 150; } ...

21. RaceCar racer = new RaceCar();

22. Car car = new RaceCar();

23 Vehicle vehicle = new RaceCar();

24 System.out.println(racer.speed() + ", " + car.speed() + ", " + vehicle. speed());

What is the result?

1. 0, 0, 0
2. 150, 60, 0
3. Compilation fails.
4. 150, 150, 150
5. An exception is thrown at runtime.

QUESTION 22

Given:

5. class Building { }

6. public class Barn extends Building {

7. public static void main(String[] args) {8.

9.

10.

11.

12.

13.

14. }

15. }

Building build1 = new Building();

Barn barn1 = new Barn();

Barn barn2 = (Barn) build1;

Object obj1 = (Object) build1;

String str1 = (String) build1;

Building build2 = (Building) barn1;

Which is true?

1. If line 10 is removed, the compilation succeeds.
2. If line 11 is removed, the compilation succeeds.
3. If line 12 is removed, the compilation succeeds.
4. If line 13 is removed, the compilation succeeds.
5. More than one line must be removed for compilation to succeed.

Correct Answer: C

QUESTION 23

A team of programmers is reviewing a proposed API for a new utility class. After some discussion, they realize that they can reduce the number of methods in the API without losing any functionality. If they implement the new design, which two OO principles will they be promoting?

1. Looser coupling
2. Tighter coupling
3. Lower cohesion
4. Higher cohesion
5. Weaker encapsulation
6. Stronger encapsulation
7. Correct Answer: A

QUESTION 24

Given:

1. class Money {
2. private String country = "Canada";
3. public String getC() { return country; }
4. }
5. class Yen extends Money {
6. public String getC() { return super.country; }
7. }
8. public class Euro extends Money {
9. public String getC(int x) { return super.getC(); }
10. public static void main(String[] args) {
11. System.out.print(new Yen().getC() + " " + new Euro().getC());
12. }
13. }
14. What is the result?
15. Canada
16. null Canada
17. Canada null
18. Canada Canada
19. Compilation fails due to an error on line 6.
20. Compilation fails due to an error on line 9.

QUESTION 25

Assuming that the serializeBanana() and the deserializeBanana() methods will correctly use Java

serialization and given:

1. import java.io.\*;
2. class Food implements Serializable {int good = 3;}
3. class Fruit extends Food {int juice = 5;}
4. public class Banana extends Fruit {
5. int yellow = 4;
6. public static void main(String [] args) {
7. Banana b = new Banana(); Banana b2 = new Banana();
8. b.serializeBanana(b); // assume correct serialization
9. b2 = b.deserializeBanana(); // assume correct
10. System.out.println("restore "+b2.yellow+ b2.juice+b2.good);
11. }
12. //
13. more Banana methods go here
14. }

What is the result?

1. restore 400
2. restore 403
3. restore 453
4. Compilation fails.
5. An exception is thrown at runtime.

QUESTION 26

Given a valid DateFormat object named df, and

16. Date d = new Date(0L);

17. String ds = "December 15, 2004";

18. //insert code here

What updates d's value with the date represented by ds?

A. 18. d = df.parse(ds);

B. 18. d = df.getDate(ds);

C. 18. try {

19. d = df.parse(ds);

20. } catch(ParseException e) { };D. 18. try {

19. d = df.getDate(ds);

20. } catch(ParseException e) { };

QUESTION 27

Given:

11. double input = 314159.26;

12. NumberFormat nf = NumberFormat.getInstance(Locale.ITALIAN);

13. String b;

14. //insert code here

Which code, inserted at line 14, sets the value of b to 314.159,26?

1. nf.parse( input );
2. nf.format( input );
3. nf.equals( input );
4. nf.parseObject( input );

QUESTION 28

Given:

1. public class TestString1 {

2. public static void main(String[] args) {

3. String str = "420";

4. str += 42;

5. System.out.print(str);

6. }

7. }

What is the output?

1. 42
2. 420
3. 462
4. 42042
5. Compilation fails.
6. An exception is thrown at runtime.

QUESTION 29

Which capability exists only in java.io.FileWriter?

A. Closing an open stream.

B. Flushing an open stream.

C. Writing to an open stream.

D. Writing a line separator to an open stream.

QUESTION 30

Given that the current directory is empty, and that the user has read and write permissions, and the

following:

1. import java.io.\*;

2. public class DOS {

3. public static void main(String[] args) {

4. File dir = new File("dir");

5. dir.mkdir();

6. File f1 = new File(dir, "f1.txt");

7. try {

8. f1.createNewFile();

9. } catch (IOException e) { ; }

10. File newDir = new File("newDir");

11. dir.renameTo(newDir);

12. }

13. }

Which statement is true?

1. Compilation fails.
2. The file system has a new empty directory named dir.
3. The file system has a new empty directory named newDir.
4. The file system has a directory named dir, containing a file f1.txt.
5. The file system has a directory named newDir, containing a file f1.txt.

QUESTION 31

Given:

static void test() throws RuntimeException {

try {

System.out.print("test ");

throw new RuntimeException();

}

catch (Exception ex) { System.out.print("exception "); }

}

public static void main(String[] args) {

try { test(); }

catch (RuntimeException ex) { System.out.print("runtime "); }

System.out.print("end ");

}

What is the result?

A. test end

B. Compilation fails.

C. test runtime endD. test exception end

E. A Throwable is thrown by main at runtime.

QUESTION 32

Given:

1. public class Score implements Comparable<Score> {

2. private int wins, losses;

3. public Score(int w, int l) { wins = w; losses = l; }

4. public int getWins() { return wins; }

5. public int getLosses() { return losses; }

6. public String toString() {

7. return "<" + wins + "," + losses + ">";

8. }

9. //

insert code here

10. }

11.

Which method will complete this class?

1. public int compareTo(Object o){/\*more code here\*/}
2. public int compareTo(Score other){/\*more code here\*/}
3. public int compare(Score s1,Score s2){/\*more code here\*/}
4. public int compare(Object o1,Object o2){/\*more code here\*/}

Section B

QUESTION 1

Given:

22. StringBuilder sb1 = new StringBuilder("123");

23. String s1 = "123";

24. // insert code here

25. System.out.println(sb1 + " " + s1);

Which code fragment, inserted at line 24, outputs "123abc 123abc"?

1. sb1.append("abc"); sb1.concat("abc");
2. sb1.append("abc"); sb1.concat("abc");
3. sb1.append("abc"); sb1.concat("abc");
4. s1.append("abc"); s1.concat("abc");
5. s1.append("abc"); s1.concat("abc");
6. s1 = s1.concat("abc"); s1 = s1.concat("abc");
7. s1 = s1 + s1.concat("abc"); s1 = s1 + s1.concat("abc");

QUESTION 2

Click the Exhibit button.

Which code, inserted at line 14, will allow this class to correctly serialize and deserialize?

1. import java.io.\*;

2. public class Foo implements Serializable {

3. public int x, y;

4. public Foo(int x, int y){

5. this.x = x; this.y = y;

6. }

7.

8. private void writeObject(ObjectOutputStream s)

9. throws IOException{

10. s.writeInt(x); s.writeInt(y);

11. }

12.

13. private void readObject(ObjectInputStream s)

14. throws IOException, ClassNotFoundException {

15.

//insert code here

16. }

17. }

1. s.defaultReadObject();
2. this = s.defaultReadObject();
3. y = s.readInt(); x = s.readInt();
4. x = s.readInt(); y = s.readInt();

QUESTION 3

Given:

1. public class LineUp {

2. public static void main(String[] args) {

3. double d = 12.345;

4. //

insert code here

5.

}

6. }

Which code fragment, inserted at line 4, produces the output | 12.345|?

1. System.out.printf("|%7d| \n", d);
2. System.out.printf("|%7f| \n", d);
3. System.out.printf("|%3.7d| \n", d);
4. System.out.printf("|%3.7f| \n", d);
5. System.out.printf("|%7.3d| \n", d);
6. System.out.printf("|%7.3f| \n", d);

QUESTION 4

Given:

11. public class Test {

12. public static void main(String [] args) {

13. int x = 5;

14. boolean b1 = true;

15. boolean b2 = false;

16.

17. if ((x == 4) && !b2 )

18. System.out.print("1 ");

19. System.out.print("2 ");

20. if ((b2 = true) && b1 )

21. System.out.print("3 ");

22. }

23. }

What is the result?

1. 2
2. 3
3. 1 2
4. 2 3
5. 1 2 3
6. Compilation fails.
7. An exception is thrown at runtime.

QUESTION 5Given:

interface Foo {}

class Alpha implements Foo {}

class Beta extends Alpha {}

class Delta extends Beta {

public static void main( String[] args ) {

Beta x = new Beta();

16. //insert code here 16

}

}

Which code, inserted at line 16, will cause a java.lang.ClassCastException?

1. Alpha a = x;
2. Foo f = (Delta)x;
3. Foo f = (Alpha)x;
4. Beta b = (Beta)(Alpha)x;

QUESTION 6

Given:

public void go() {

String o = "";

z:

for(int x = 0; x < 3; x++) {

for(int y = 0; y < 2; y++) {

if(x==1) break;

if(x==2 && y==1) break z;

o = o + x + y;

}

}

System.out.println(o);

}

What is the result when the go() method is invoked?

1. 00
2. 0001
3. 000120
4. 00012021
5. Compilation fails.
6. An exception is thrown at runtime.

QUESTION 7

Given:

try {

//some code here line 34

} catch (NullPointerException e1) {

System.out.print("a");

} catch (Exception e2) {

System.out.print("b");} finally {

System.out.print("c");

}

If some sort of exception is thrown at line 34, which output is possible?

1. a
2. b
3. c
4. ac
5. abc

QUESTION 8

Given:

//some code here line 31

try {

//some code here line 33

} catch (NullPointerException e1) {

//some code here line 35

} finally {

//some code here line 37

}

Under which three circumstances will the code on line 37 be executed? (Choose three.)

1. The instance gets garbage collected.
2. The code on line 33 throws an exception.
3. The code on line 35 throws an exception.
4. The code on line 31 throws an exception.
5. The code on line 33 executes successfully.

QUESTION 9

Given:

int x = 0;

int y = 10;

do {

y--;

++x;

} while (x < 5);

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

What is the result?

1. 5,6
2. 5,5
3. 6,5
4. 6,6

QUESTION 10

Given:

public class Donkey2 {

public static void main(String[] args) {

boolean assertsOn = true;

assert (assertsOn) : assertsOn = true;

if(assertsOn) {

System.out.println("assert is on");

}

}

}

If class Donkey is invoked twice, the first time without assertions enabled, and the second time with

assertions enabled, what are the results?

A. no output

B. no output

assert is on

C. assert is on

D. no output

An AssertionError is thrown.

E. assert is on

An AssertionError is thrown.

QUESTION 12

Given:

Float pi = new Float(3.14f);

if (pi > 3) {

System.out.print("pi is bigger than 3. ");

}

else {

System.out.print("pi is not bigger than 3. ");

}

finally {

System.out.println("Have a nice day.");

}

What is the result?

1. Compilation fails.
2. pi is bigger than 3.
3. An exception occurs at runtime.
4. pi is bigger than 3. Have a nice day.
5. pi is not bigger than 3. Have a nice day.

QUESTION 13

Given:

1. public class Boxer1{

2. Integer i;

3. int x;

4. public Boxer1(int y) {

5. x = i+y;

6. System.out.println(x);

7. }

8. public static void main(String[] args) {

9. new Boxer1(new Integer(4));

10. }

11. }

What is the result?

1. The value "4" is printed at the command line.
2. Compilation fails because of an error in line 5.
3. Compilation fails because of an error in line 9.
4. A NullPointerException occurs at runtime.
5. A NumberFormatException occurs at runtime.
6. An IllegalStateException occurs at runtime.

QUESTION 14

Given:

1. public class Person {

2. private String name;

3. public Person(String name) { this.name = name; }

4. public boolean equals(Person p) {

5. return p.name.equals(this.name);

6. }

7. }

Which statement is true?

A. The equals method does NOT properly override the Object.equals method.

B. Compilation fails because the private attribute p.name cannot be accessed in line 5.

C. To work correctly with hash-based data structures, this class must also implement the hashCode

method.

D. When adding Person objects to a java.util.Set collection, the equals method in line 4 will prevent

duplicates.

QUESTION 15

Which two statements are true about the hashCode method? (Choose two.)

A. The hashCode method for a given class can be used to test for object equality and object inequality for that class.

B. The hashCode method is used by the java.util.SortedSet collection class to order the elements within that set.

C. The hashCode method for a given class can be used to test for object inequality, but NOT object

equality, for that class.

D. The only important characteristic of the values returned by a hashCode method is that the distribution of values must follow a Gaussian distribution.

E. The hashCode method is used by the java.util.HashSet collection class to group the elements within that set into hash buckets for swift retrieval.

QUESTION 16

Given a pre-generics implementation of a method:11. public static int sum(List list) {

12. int sum = 0;

13. for ( Iterator iter = list.iterator(); iter.hasNext(); ) {

14. int i = ((Integer)iter.next()).intValue();

15. sum += i;

16. }

17. return sum;

18. }

What three changes allow the class to be used with generics and avoid an unchecked warning? (Choose three.)

1. Remove line 14.
2. Replace line 14 with "int i = iter.next();".
3. Replace line 13 with "for (int i : intList) {".
4. Replace line 13 with "for (Iterator iter : intList) {".
5. Replace the method declaration with "sum(List<int> intList)".
6. Replace the method declaration with "sum(List<Integer> intList)".

QUESTION 17

Given:

23. Object [] myObjects = {

24. new Integer(12),

25. new String("foo"),

26. new Integer(5),

27. new Boolean(true)

28. };

29. Arrays.sort(myObjects);

30. for(int i=0; i<myObjects.length; i++) {

31. System.out.print(myObjects[i].toString());

32. System.out.print(" ");

33. }

What is the result?

Compilation fails due to an error in line 23.

Compilation fails due to an error in line 29.

A ClassCastException occurs in line 29.

A ClassCastException occurs in line 31.

The value of all four objects prints in natural order.

QUESTION 19

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?

1. /test/Chess.class
2. /home/bob/Chess.class
3. /test/games/Chess.class
4. /usr/lib/games/Chess.class
5. /home/bob/games/Chess.class
6. inside jarfile /opt/java/lib/Games.jar (with a correct manifest)
7. inside jarfile /home/bob/downloads/Games.jar (with a correct manifest)

QUESTION 20

Given the following directory structure:

bigProject

|--source

| |--Utils.java

|

|--classes

|--

And the following command line invocation:

javac -d classes source/Utils.javaAssume the current directory is bigProject, what is the result?

1. If the compile is successful, Utils.class is added to the source directory.
2. The compiler returns an invalid flag error.
3. If the compile is successful, Utils.class is added to the classes directory.
4. If the compile is successful, Utils.class is added to the bigProject directory.

QUESTION 21

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.)

1. java MainClass if run from the /apps directory
2. java com.company.application.MainClass if run from the /apps directory
3. java -classpath /apps com.company.application.MainClass if run from any directory
4. java -classpath . MainClass if run from the /apps/com/company/application directory
5. java -classpath /apps/com/company/application:. MainClass if run from the /apps directory
6. java com.company.application.MainClass if run from the /apps/com/company/application directory

QUESTION 22

Given:

interface DoStuff2 {

float getRange(int low, int high);

}

interface DoMore {

float getAvg(int a, int b, int c);

}

abstract class DoAbstract implements DoStuff2, DoMore {

}

class DoStuff implements DoStuff2 {

public float getRange(int x, int y) {

return 3.14f;

}

}

interface DoAll extends DoMore {

float getAvg(int a, int b, int c, int d);

}

What is the result?

1. The file will compile without error.
2. Compilation fails. Only line 7 contains an error.
3. Compilation fails. Only line 12 contains an error.
4. Compilation fails. Only line 13 contains an error.
5. Compilation fails. Only lines 7 and 12 contain errors.
6. Compilation fails. Only lines 7 and 13 contain errors.
7. Compilation fails. Lines 7, 12, and 13 contain errors.