JAVA MIXED MCQ’S

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.

Correct Answer: DE

**Explanation/Reference:**

End of method.

run.

Exception in thread "Thread-0" java.lang.RuntimeException: Problem

at Threads2.run(Threads2.java:5)

at java.lang.Thread.run(Unknown Source)

**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().

**Correct Answer:** AF

**QUESTION 3**

Given:

**void** waitForSignal() {

Object obj = **new** Object();

**synchronized** (Thread.currentThread()) {

obj.wait();

obj.notify();

}

}

Which statement is true?

A. This code can throw an InterruptedException.

B. This code can throw an IllegalMonitorStateException.

C. This code can throw a TimeoutException after ten minutes.

D. Reversing the order of obj.wait() and obj.notify() might cause this method to complete normally.

E. A call to notify() or notifyAll() from another thread might cause this method to complete normally.

F. This code does NOT compile unless "obj.wait()" is replaced with "((Thread) obj).wait()".

**Correct Answer:** A

**Section: All**

**Explanation:**Threads2.java:15: unreported.exception java.lang.InterruptedException; mustbe caught or declared to be thrown

obj.wait();

^

1 error

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.}

A. 4

B. 5

C. 8

D. 9

E. Compilation fails.

F. An exception is thrown at runtime.

G. It is impossible to determine for certain.

**Correct Answer:** D

**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?

A. The output could be 5-1 6-1 6-2 5-2

B. The output could be 6-1 6-2 5-1 5-2

C. The output could be 6-1 5-2 6-2 5-1

D. The output could be 6-1 6-2 5-1 7-1

**Correct Answer:** B

**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?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "foo".

D. The code executes normally, but nothing is printed.

**Correct Answer:** B

**Explanation**

fooException in thread "main" java.lang.IllegalThreadStateException

at java.lang.Thread.start(Unknown Source)

at p6.Threads4.go(Threads4.java:15)

at p6.Threads4.main(Threads4.java:5)

**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 \*/}

}

**Correct Answer:** BE

**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?

A. hi hi

B. hi world

C. world world

D. Compilation fails.

E. An exception is thrown at runtime.

**Correct Answer:** D

**Explanation/Reference:**

The method go(String[], int) in the type Barn is not applicable for the arguments (String, int)

The variable argument type String of the method go must be the last parameter.

**QUESTION 9**

Which Man class properly represents the relationship "Man has a best friend who is a Dog"?

A. **class** Man **extends** Dog { }

B. **class** Man **implements** Dog { }

C. **class** Man { **private** BestFriend dog; }

D. **class** Man { **private** Dog bestFriend; }

E. **class** Man { **private** Dog<bestFriend>; }

F. **class** Man { **private** BestFriend<dog>; }

**Correct Answer:** D

**QUESTION 10**

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?

A. any class

B. only the Target class

C. any class in the test package

D. any class that extends Target

**Correct Answer:** C

**QUESTION 11**

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?

A. 0, 0, 0

B. 150, 60, 0

C. Compilation fails.

D. 150, 150, 150

E. An exception is thrown at runtime.

**Correct Answer:** D

**QUESTION 12**

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?

A. Looser coupling

B. Tighter coupling

C. Lower cohesion

D. Higher cohesion

E. Weaker encapsulation

F. Stronger encapsulation

**Correct Answer:** A

**QUESTION 13**

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

serialization and given:

**13. import** java.io.\*;

**14. class** Food **implements** Serializable {**int** good = 3;}

**15. class** Fruit **extends** Food {**int** juice = 5;}

**16. public class** Banana **extends** Fruit {

17. **int** yellow = 4;

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

19. Banana b = **new** Banana(); Banana b2 = **new** Banana();

20. b.serializeBanana(b); // assume correct serialization

21. b2 = b.deserializeBanana(); // assume correct

22. System.out.println("restore "+b2.yellow+ b2.juice+b2.good);

24. }

25. // more Banana methods go here

50. }

What is the result?

A. restore 400

B. restore 403

C. restore 453

D. Compilation fails.

E. An exception is thrown at runtime.

**Correct Answer:** C

**QUESTION 14**

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?

A. 42

B. 420

C. 462

D. 42042

E. Compilation fails.

F. An exception is thrown at runtime.

**Correct Answer:** D

**QUESTION 15**

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.

**Correct Answer:** D

**QUESTION 16**

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?

A. Compilation fails.

B. The file system has a new empty directory named dir.

C. The file system has a new empty directory named newDir.

D. The file system has a directory named dir, containing a file f1.txt.

E. The file system has a directory named newDir, containing a file f1.txt.

**Correct Answer:** E

**QUESTION 17**

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 end

D. test exception end

E. A Throwable is thrown by main at runtime.

**Correct Answer:** D

**QUESTION 18**

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"?

A. sb1.append("abc"); s1.append("abc");

B. sb1.append("abc"); s1.concat("abc");

C. sb1.concat("abc"); s1.append("abc");

D. sb1.concat("abc"); s1.concat("abc");

E. sb1.append("abc"); s1 = s1.concat("abc");

F. sb1.concat("abc"); s1 = s1.concat("abc");

G. sb1.append("abc"); s1 = s1 + s1.concat("abc");

H. sb1.concat("abc"); s1 = s1 + s1.concat("abc");

**Correct Answer:** E

**QUESTION 19**

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?

A. 2

B. 3

C. 1 2

D. 2 3

E. 1 2 3

F. Compilation fails.

G. An exception is thrown at runtime.

**Correct Answer:** D

**QUESTION 20**

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?

A. 00

B. 0001

C. 000120

D. 00012021

E. Compilation fails.

F. An exception is thrown at runtime.

**Correct Answer:** C

**QUESTION 21**

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?

A. a

B. b

C. c

D. ac

E. abc

**Correct Answer:** D

**QUESTION 22**

Given:

**int** x = 0;

**int** y = 10;

**do** {

y--;

++x;

} **while** (x < 5);

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

What is the result?

A. 5,6

B. 5,5

C. 6,5

D. 6,6

**Correct Answer:** B

**QUESTION 23**

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.

**Correct Answer:** A

**QUESTION 24**

Given the following directory structure:

bigProject

|--source

| |--Utils.java

| |--classes

|--

And the following command line invocation:

javac -d classes source/Utils.java

Assume the current directory is bigProject, what is the result?

A. If the compile is successful, Utils.class is added to the source directory.

B. The compiler returns an invalid flag error.

C. If the compile is successful, Utils.class is added to the classes directory.

D. If the compile is successful, Utils.class is added to the bigProject directory.

**Correct Answer:** C

**QUESTION 25**

Which statement is true?

A. A class's finalize() method CANNOT be invoked explicitly.

B. super.finalize() is called implicitly by any overriding finalize() method.

C. The finalize() method for a given object is called no more than once by the garbage collector.

D. The order in which finalize() is called on two objects is based on the order in which the two objects

became finalizable.

**Correct Answer:** C

**QUESTION 26**

A company has a business application that provides its users with many different reports:receivables reports, payables reports, revenue projects, and so on. The company has just purchased some new, state-of-the-art, wireless printers, and a programmer has been assigned the task of enhancing all of the reports to use not only the company's old printers, but the new wireless printers as well. When the programmer starts looking into the application, the programmer discovers that because of the design of the application, it is necessary to make changes to each report to support the new printers.Which two design

concepts most likely explain this situation? (Choose two.)

A. Inheritance

B. Low cohesion

C. Tight coupling

D. High cohesion

E. Loose coupling

F. Object immutability

**Correct Answer:** BC

**QUESTION 27**

A company that makes Computer Assisted Design (CAD) software has, within its application, some utility classes that are used to perform 3D rendering tasks. The company's chief scientist has just improved the

performance of one of the utility classes' key rendering algorithms, and has assigned a programmer to replace the old algorithm with the new algorithm. When the programmer begins researching the utility classes, she is happy to discover that the algorithm to be replaced exists in only one class. The programmer reviews that class's API, and replaces the old algorithm with the new algorithm, being careful that her changes adhere strictly to the class's API. Once testing has begun, the programmer discovers that other classes that use the class she changed are no longer working properly. What design flaw is most likely the cause of these new bugs?

A. Inheritance

B. Tight coupling

C. Low cohesion

D. High cohesion

E. Loose coupling

F. Object immutability

**Correct Answer:** B

**QUESTION 28**

Given:

10. **interface** Jumper { **public void** jump(); }

...

20. **class** Animal {}

...

30. **class** Dog **extends** Animal {

31. Tail tail;

32. }

...

40. **class** Beagle **extends** Dog **implements** Jumper{

41. **public void** jump() {}

42. }

...

50. **class** Cat **implements** Jumper{

51. **public void** jump() {}

52. }

Which three are true? (Choose three.)

A. Cat is-a Animal

B. Cat is-a Jumper

C. Dog is-a Animal

D. Dog is-a Jumper

E. Cat has-a Animal

F. Beagle has-a Tail

G. Beagle has-a Jumper

**Correct Answer:** BCF

**QUESTION 29**

Given:

1. **import** java.util.\*;

2. **public class** WrappedString {

3. **private** String s;

4. **public** WrappedString(String s) { **this**.s = s; }

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

6. HashSet<Object> hs = **new** HashSet<Object>();

7. WrappedString ws1 = **new** WrappedString("aardvark");

8. WrappedString ws2 = **new** WrappedString("aardvark");

9. String s1 = **new** String("aardvark");

10. String s2 = **new** String("aardvark");

11. hs.add(ws1); hs.add(ws2); hs.add(s1); hs.add(s2);

12. System.out.println(hs.size()); } }

What is the result?

A. 0

B. 1

C. 2

D. 3

E. 4

F. Compilation fails.

G. An exception is thrown at runtime.

**Correct Answer:** D

**QUESTION 30**

Given:

**enum** Example { ONE, TWO, THREE }

Which statement is true?

A. The expressions (ONE == ONE) and ONE.equals(ONE) are both guaranteed to be true.

B. The expression (ONE < TWO) is guaranteed to be true and ONE.compareTo(TWO) is guaranteed to be

less than one.

C. The Example values cannot be used in a raw java.util.HashMap; instead, the programmer must use a

java.util.EnumMap.

D. The Example values can be used in a java.util.SortedSet, but the set will NOT be sorted because

enumerated types do NOT implement java.lang.Comparable.

**Correct Answer:** A

**QUESTION 31**

Given:

**import** java.util.\*;

**public class** Quest {

**public static void** main(String[] args) {

String[] colors = {"blue", "red", "green", "yellow", "orange"};

Arrays.sort(colors);

**int** s2 = Arrays.binarySearch(colors, "orange");

**int** s3 = Arrays.binarySearch(colors, "violet");

System.out.println(s2 + " " + s3);

}

}

What is the result?

A. 2 -1

B. 2 -4

C. 2 -5

D. 3 -1

E. 3 -4

F. 3 -5

G. Compilation fails.

H. An exception is thrown at runtime.

**Correct Answer:** C

**QUESTION 32**

Which two statements are true? (Choose two.)

A. It is possible to synchronize static methods.

B. When a thread has yielded as a result of yield(), it releases its locks.

C. When a thread is sleeping as a result of sleep(), it releases its locks.

D. The Object.wait() method can be invoked only from a synchronized context.

E. The Thread.sleep() method can be invoked only from a synchronized context.

F. When the thread scheduler receives a notify() request, and notifies a thread, that thread immediately

releases its lock.

**Correct Answer:** AD

**QUESTION 33**

Given:

**public class** TestOne **implements** Runnable {

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

Thread t = **new** Thread(**new** TestOne());

t.start();

System.out.print("Started");

t.join();

System.out.print("Complete");

}

**public void** run() {

**for** (**int** i = 0; i < 4; i++) {

System.out.print(i);

}

}

}

What can be a result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes and prints "StartedComplete".

D. The code executes and prints "StartedComplete0123".

E. The code executes and prints "Started0123Complete".

**Correct Answer:** E

**QUESTION 34**

Given that t1 is a reference to a live thread, which is true?

A. The Thread.sleep() method can take t1 as an argument.

B. The Object.notify() method can take t1 as an argument.

C. The Thread.yield() method can take t1 as an argument.

D. The Thread.setPriority() method can take t1 as an argument.

E. The Object.notify() method arbitrarily chooses which thread to notify.

**Correct Answer:** E

**QUESTION 35**

Given:

Runnable r = **new** Runnable() {

**public void** run() {

System.out.print("Cat");

}

};

Thread t = **new** Thread(r) {

**public void** run() {

System.out.print("Dog");

}

};

t.start();

What is the result?

A. Cat

B. Dog

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

**Correct Answer:** B

**QUESTION 36**

Given:

1. **public class** Threads5 {

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

3. **new** Thread(**new** Runnable() {

4. **public void** run() {

5. System.out.print("bar");

6. }}).start();

7. }

8. }

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "bar".

D. The code executes normally, but nothing prints.

**Correct Answer:** C

**QUESTION 37**

Given:

**class** One {

**void** foo() { }

}

**class** Two **extends** One {

14. // insert method here

}

Which three methods, inserted individually at line 14, will correctly complete class Two? (Choose three.)

A. int foo() { /\* more code here \*/ }

B. void foo() { /\* more code here \*/ }

C. public void foo() { /\* more code here \*/ }

D. private void foo() { /\* more code here \*/ }

E. protected void foo() { /\* more code here \*/ }

**Correct Answer:** BCE

**QUESTION 38**

Given:

1. **public class** A {

2. **public void** doit() {

3. }

4. **public** String doit() {

5. **return** "a";

6. }

7. **public double** doit(**int** x) {

8. **return** 1.0;

9. }

10.}

What is the result?

A. An exception is thrown at runtime.

B. Compilation fails because of an error in line 7.

C. Compilation fails because of an error in line 4.

D. Compilation succeeds and no runtime errors with class A occur.

**Correct Answer:** C

**QUESTION 38**

Given

**11. public interface** Status {

12. /\* insert code here \*/ **int** MY\_VALUE = 10;

13. }

Which three are valid on line 12? (Choose three.)

A. final

B. static

C. native

D. public

E. private

F. abstract

G. protected

**Correct Answer:** ABD

**QUESTION 39**

Given:

**public static** Collection get() {

Collection sorted = **new** LinkedList();

sorted.add("B"); sorted.add("C"); sorted.add("A");

**return** sorted;

}

**public static void** main(String[] args) {

**for** (Object obj: get()) {

System.out.print(obj + ", ");

}

}

What is the result?

A. A, B, C,

B. B, C, A,

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

**Correct Answer:** B

**QUESTION 40**

Given:

1. **public class** TestString3 {

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

3. // insert code here

4.

5. System.*out*.println(s);

6. }

7. }

Which two code fragments, inserted independently at line 3, generate the output 4247? (Choose two.)

A. String s = "123456789";

s = (s-"123").replace(1,3,"24") - "89";

B. StringBuffer s = new StringBuffer("123456789");

C. s.delete(0,3).replace(1,3,"24").delete(4,6);

D. StringBuffer s = new StringBuffer("123456789");

E. substring(3,6).delete(1,3).insert(1, "24");

F. StringBuilder s = new StringBuilder("123456789");

G. substring(3,6).delete(1,2).insert(1, "24");

H. StringBuilder s = new StringBuilder("123456789");

I. delete(0,3).delete(1,3).delete(2,5).insert(1, "24");

**Correct Answer:** BC

**QUESTION 41**

Which four statements are true? (Choose four.)

A. Has-a relationships should never be encapsulated.

B. Has-a relationships should be implemented using inheritance.

C. Has-a relationships can be implemented using instance variables.

D. Is-a relationships can be implemented using the extends keyword.

E. Is-a relationships can be implemented using the implements keyword.

F. The relationship between Movie and Actress is an example of an is-a relationship.

G. An array or a collection can be used to implement a one-to-many has-a relationship.

**Correct Answer:** CDEG

**QUESTION 42**

Given:

**public class** Hello {

String title;

**int** value;

**public** Hello() {

title += " World";

}

**public** Hello(**int** value) {

**this**.value = value;

title = "Hello";

Hello();

}

}

and:

Hello c = new Hello(5);

System.out.println(c.title);

What is the result?

A. Hello

B. Hello World

C. Compilation fails.

D. Hello World 5

E. The code runs with no output.

F. An exception is thrown at runtime.

**Correct Answer:** C