1.Given:

11. public interface Word { boolean isSpelled(String w); }

12.

13. abstract class Verb1 implements Word { boolean isSpelled() { } } 14.

15. abstract class Verb2 implements Word { boolean isSpelled() { return true; }} 16.

17. abstract class Verb3 implements Word {

18. boolean isSpelled(String w) { return true; }

19. }

What is the result?

A. Compilation succeeds.

B. Compilation fails due only to an error on line 13.

C. Compilation fails due only to an error on line 15.

D. Compilation fails due only to an error on line 18.

E. Compilation fails due only to errors on lines 13 and 15.

F. Compilation fails due only to errors on lines 13 and 18.

G. Compilation fails due only to errors on lines 15 and 18.

2.Given:

3. public class Eggs {

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

5. int[][] basket1 = new int[2][];

6. basket1[0] = new int[3];

7. basket1[0] = {1,2,3};

8.

9. int[][] basket2 = new int[2][];

10. basket2[0] = new int[] {1,2,3};

11. basket2[0][1] = 7;

12. System.out.println(basket1[0][1] + " " + basket2[0][1]); 13. }

14. }

What is the result?

A. 1 1

B. 2 2

C. 1 7

D. 2 7

E. Compilation fails

F. An exception is thrown at runtime

3.Given:

1. class CopyArray {

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

3. int[] x = {1, 2 ,3};

4. // insert code here

5. }

6. }

Which two, inserted independently at line 4, will compile? (Choose two.)

A. int[] y1 = x;

B. int[] y2; y2 = x;

C. int[] y3 = x.copy();

D. int[] y4; for(int z : x) { y4[z] = x[z]; }

4.Given:

1. class Banana {

2. int x = 1;

3. public static void main(String [] args) { 4. int x = 2;

5. Banana b = new Banana();

6. b.go();

7. }

8. { x += x; }

9. void go() {

10. ++x;

11. System.out.println(x);

12. }

13. }

What is the result?

A. 1

B. 2

C. 3

D. 5

E. Compilation fails. 

5.Given:

1. class SuperFoo {

2. SuperFoo doStuff(int x) {

3. return new SuperFoo();

4. }

5. }

6.

7. class Foo extends SuperFoo {

8. // insert code here

9. }

And four declarations:

I. Foo doStuff(int x) { return new Foo(); }

II. Foo doStuff(int x) { return new SuperFoo(); }

III. SuperFoo doStuff(int x) { return new Foo(); }

IV. SuperFoo doStuff(int y) { return new SuperFoo(); }

Which, inserted independently at line 8, will compile?

A. Only I

B. Only IV

C. Only I and III

D. Only I, II, and III

E. Only I, III, and IV

F. All four declarations will compile

6.Given:

1. class FWD {

2. int doMud(int x) { return 1; }

3. }

4. class Subaru extends FWD {

5. int doMud(int... y) { return 2; }

6. int doMud(int z) { return 3; }

7. }

8. class Race {

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

10. int s = new Subaru().doMud(7);

11. System.out.println(s);

12. }

13. }

What is the result?

A. 1

B. 2

C. 3

D. 7

E. Compilation fails

F. The output is NOT predictable

7.Given:

1. class HorseRadish {

2. // insert code here

3. protected HorseRadish(int x) {

4. System.out.println("bok choy");

5. }

6. }

7. class Wasabi extends HorseRadish {

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

9. Wasabi w = new Wasabi();

10. }

11. }

Which two, inserted independently at line 2, will allow the code to compile and produce the output "bok choy"? (Choose two.)

A. // just a comment

B. protected HorseRadish() { }

C. protected HorseRadish() { this(42);}

D. protected HorseRadish() { new HorseRadish(42);}

8.Given:

4. public class FreeRange {

5. public static void main(String[] args) { 6. int x = 7, y = 8;

7. if(x < y)

8. if(x+2 > y)

9. if(y < x) ;

10. else if(!false)

11. System.out.print("inner ");

12. else if(true)

13. System.out.print("middle ");

14. }

15. }

What is the result?

A. inner

B. middle

C. inner middle

D. middle inner

E. Compilation fails

F. An exception is thrown at runtime



9.Given:

5. public class Buddy {

6. public static void main(String[] args) { 7. def:

8. for(short s = 1; s < 7; s++) { 9. if(s == 5) break def;

10. if(s == 2) continue;

11. System.out.print(s + ".");

12. }

13. }

14. }

What is the result?

A. 1.

B. 1.2.

C. 1.3.4.

D. 1.2.3.4.

E. 1.3.4.5.6.

F. 1.2.3.4.5.6.

G. Compilation fails.

10.Given:

1. class Parser extends Utils {

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

3. try {

System.out.print(new Parser().getInt("42"));

4. } catch (Exception e) {

5. System.out.println("Exc"); }

6. }

7. int getInt(String arg) throws Exception {

8. return Integer.parseInt(arg);

9. }

10. }

11. class Utils {

12. int getInt(String arg) { return 42; }

13. }

What is the result?

A. 42

B. Exc

C. 42Exc

D. Compilation fails

E. An exception is thrown at runtime



11.Given:

1. class Propeller2 {

2. public static void main(String [] args) // add code here? 3. { new Propeller2().topGo(); }

4.

5. void topGo() // add code here?

6. { middleGo(); }

7.

8. void middleGo() // add code here?

9. {

go();

System.out.println("late middle "); }

10.

11. void go() // add code here?

12. { throw new Exception(); }

13. }

For the code to compile, on which lines does the declaration throws Exception need to be added?

A. Only line 11

B. Lines 8 and 11

C. Lines 5, 8, and 11

D. Lines 2, 5, 8, and 11

E. No combination of these additions will allow the code to compile

12.Given:

1. class StringTest {

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

3. String s = null;

4. try {

5. s.trim();

6. } catch (Exception e) {

7. System.out.println("exc");

8. }

9. s.trim();

10. }

11. }

What is the result?

A. Compilation fails

B. The code runs with no output

C. An exception is thrown with no other output

D. exc then an exception is thrown

E. exc then the code completes without exception

13.Given:

1. class Dog { }

2. class BorderCollie extends Dog { }

3. class Kennel {

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

5. BorderCollie b = new BorderCollie();

6. Dog d = (Dog)b;

7. }

8. }

What is the result?

A. Compilation fails

B. The code runs with no output

C. A java.lang.ClassCastException is thrown

D. A java.lang.IllegalStateException is thrown E. A java.lang.ExceptionInInitializationError is thrown

14.Given:

1. class Mutate {

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

3. StringBuilder s = new StringBuilder("012345678 ");

4. if (s.length() == 10)

5. s.insert(10, "abcdef");

6. s.delete(3,8);

7. System.out.println(s.indexOf("c"));

8. }

9. }

What is the result?

A. -1

B. 5

C. 6

D. 7

E. Compilation fails

F. An exception is thrown at runtime

15.Which two statements comparing java.lang.StringBuilder to java.lang.StringBuffer are true? (Choose two.)

A. Both classes have a lastIndexOf method

B. Both classes' key methods are synchronized

C. Only StringBuffer has a lastIndexOf method

D. Only StringBuilder has a lastIndexOf method

E. Only StringBuffer's key methods are synchronized

F. Only StringBuilder's key methods are synchronized

16.Given:

1. class Alpha { void m1() {} }

2. class Beta extends Alpha { void m2() { } }

3. class Gamma extends Beta { }

4.

5. class GreekTest {

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

7. Alpha [] a = {new Alpha(), new Beta(), new Gamma() }; 8. for(Alpha a2 : a) {

9. a2.m1();

10. if (a2 instanceof Beta || a2 instanceof Gamma)

11. // insert code here

12. }

13. }

14. }

Which code, inserted at line 11, will compile, but cause an exception to be thrown at runtime? A. a2.m2();

B. ((Beta)a2).m2();

C. ((Alpha)a2).m2();

D. ((Gamma)a2).m2(); 

17.Given:

7. public class Redwood extends Tree {

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

9. Tree t1 = new Tree();

10. Tree t2 = new Redwood();

11. Redwood r1 = new Redwood();

12. // insert code here

13. }

14. }

15. class Tree { }

And the four fragments:

I. Redwood r2 = t2;

II. Redwood r3 = (Redwood) t2;

III. Tree t3 = t2;

IV. Tree t4 = (Tree) t2;

Which fragments, inserted independently at line 12, will compile?

A. Only fragments I and II

B. Only fragments III and IV

C. Only fragments I, II, and IV

D. Only fragments I, II, and III

E. Only fragments II, III, and IV

F. All four of the fragments will compile

18.Given:

1. class Book {

2. private final void read() { System.out.print("book "); } 3. }

4. class Page extends Book {

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

6. new Page().read();

7. new Book().read();

8. }

9. private final void read() { System.out.print("page "); } 10. }

What is the result?

A. book book

B. book page

C. page book

D. page page

E. Compilation fails

F. An exception is thrown at runtime



19.Given:

1. class Tree {

2. private static String tree = "tree ";

3. String getTree() { return tree; }

4. }

5. class Elm extends Tree {

6. private static String tree = "elm ";

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

8. new Elm().go(new Tree());

9. }

10. void go(Tree t) { tree elm this = elm

11. String s = t.getTree() + Elm.tree + this.!!tree + (new

Tree

Elm().getTree());

12. System.out.println(s);

13. } }

What is the result?

A. elm elm elm elm

B. tree elm elm elm

C. tree elm tree elm

D. tree elm elm tree

E. Compilation fails

F. An exception is thrown at runtime

20.Given:

4. class Chemical {

5. int ph() { return 7; }

6. }

7. public class Acid !exetends Ch{

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

9. new Acid().go();

10. }

11. void go() {

12. System.out.println(ph() + " " + super.ph() + " " + getPh()); 13. }

14. int getPh() { return 4; }

15. int ph() { return 3; }

15. }

What is the result?

A. 3 7 4

B. 4 7 4

C. 7 7 4

D. Compilation fails

E. An exception is thrown at runtime

21.Given:

1. class Over {

2. int doIt(long x) { return 3; }

3. }

4.

5. class Under extends Over {

6. // insert code here

7. }

And the four methods:

I. short doIt(int y) { return 4; }

II. int doIt(long x, long y) { return 4; }

III. private int doIt(short y) { return 4; }

IV. protected int doIt(long x) { return 4; }

Which methods, inserted independently at line 6, will compile?

A. Only methods I and II

B. Only methods II and IV

C. Only methods I and IV

D. Only methods I, III, and IV

E. Only methods I, II, and III

F. All of the above methods will compile



22.Given:

3. class YaYa { Yum yum; }

4. interface Yum { void yip(); }

5. class Fizz extends YaYa implements Yum { 6. public void yip() { }

7. }

8. interface Zoom extends Yum { }

Which two are true? (Choose two.)

A. Class Fizz is-a Yum

B. Class YaYa is-a Yum

C. Class Fizz has-a Yum

D. Class Fizz has-a YaYa

E. Interface Yum has-a yip

F. Interface Zoom has-a yip

23.Given:

3. public class Tiger {

4. public static void main(String[] args) { 5. short stop = 7;

6. String s = "3";

7. System.out.print(s + stop + " "); 8. System.out.print(stop + s + " "); 9. System.out.print(stop + 1 + s); 10. }

11. }

What is the result?

A. 10 10 11

B. 37 10 11

C. 37 73 83

D. 37 73 713

E. Compilation fails

F. An exception is thrown at runtime

24.Given: !!!!!

3. import java.util.\*;

4. public class ToDo {

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

6. String[] dogs = {"fido" "clover" "gus" "aiko"};

7. List dogList = Arrays.asList(dogs);

8. dogList.add("spot");

9. dogs[0] = "fluffy";

10. System.out.println(dogList);

11. for(String s: dogs) System.out.print(s + ", "); 12. }

13. }

What is the result?

A. [fluffy, clover, gus, aiko]

fluffy, clover, gus, aiko,

B. [fluffy, clover, gus, aiko]

fluffy, clover, gus, aiko, spot,

C. fluffy, clover, gus, aiko, spot]

fluffy, clover, gus, aiko,

D. [fluffy, clover, gus, aiko, spot]

fluffy, clover, gus, aiko, spot,

E. Compilation fails

F. An exception is thrown at runtime 

25.Given:

3. interface MyInterface {

4. static long boat = 7L;

5. long myMethod(long x);

6. }

7. public class TestIface implements MyInterface { 8. public static void main(String[] args) { 9. new TestIface().myMethod(6L);

10. }

11. long myMethod(long x) {

12. System.out.println( ((++x \* boat) - (--x + 1)) ); 13. return 42L;

14. }

15. }

What is the result?

A. 40

B. 41

C. 42

D. 43

E. Compilation fails

F. An exception is thrown at runtime

26.Given two files:

1. package x;

2. public class X {

3. public static void doX() { System.out.print("doX "); } 4. }

And:

1. class Find {

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

3. // insert code here

4. }

5. }

Which two, inserted independently at line 3 in class Find, will compile and produce the output "doX"? (Choose two.)

A. doX();

B. X.doX();

C. x.X.doX();

D. X myX = new X(); myX.doX();

E. x.X myX = new x.X(); myX.doX();







27.Given:

3. class Bottle { }

4. public class Recycle {

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

6. Bottle b1 = new Bottle();

7. Bottle b2 = b1;

8. b1 = new Bottle();

9. Bottle b3 = b2;

10. b2 = null;

11. b1 = b3;

12. // do stuff

13. }

14. }

What is true about objects being eligible for the garbage collector (GC), when line 12 is reached during execution?

A. No Bottle objects are eligible for GC

B. Only the first Bottle object created is eligible for GC

C. Only the second Bottle object created is eligible for GC

D. The first two Bottle objects created are eligible for GC

E. It is impossible to determine which objects are eligible for the GC

28.Given the two source files:

1. package com.sun;

2. public class PkgAccess {

3. public static int tiger = 1414;

4. }

And:

1. import static com.sun.PkgAccess.\*; 2.

3. public class PkgAccess2 {

4.

5. int x1 = PkgAccess.tiger;

6. int x2 = tiger;

7. int x3 = com.sun.PkgAccess.tiger; 8. int x4 = sun.PkgAccess.tiger;

9. }

Which two are true? (Choose two.)

A. The PkgAccess2 class compiles

B. Compilation fails due to an error on line 5

C. Compilation fails due to an error on line 6

D. Compilation fails due to an error on line 7

E. Compilation fails due to an error on line 8

F. The PkgAccess and PkgAccess2 classes both compile 

29.Given:

2. interface Picker { Integer tomato = 57;}

3.

4. interface Grinner {

5. int doStuff();

6. }

7.

8. interface Sinner extends Picker, Grinner {

9. int doStuff(boolean maybe);

10. int doStuff();

11. }

What is the result?

A. All three interfaces will compile

B. Compilation fails due to multiple errors

C. Compilation fails due only to an error on line 2 D. Compilation fails due only to an error on line 8 E. Compilation fails due only to an error on line 9 F. Compilation fails due only to an error on line 10

30.Given:

3. public class Buckets {

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

5. int[] x = {1,2,3};

6. int[] y = {4,5,6};

7. int[] z = {7,8,9};

8. go(x, y, z);

9. go(y, z);

10. go(z);

11. }

12. static void go(int[] a, int[]... b) {

13. System.out.print(a[1] + " ");

14. }

15. }

What is the result?

A. 1 2 3

B. 1 4 7

C. 2 5 8

D. 4 5 6

E. Compilation fails

F. An exception is thrown at runtime



31.Given:

1. class DoStuff {

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

3. doIt(1);

4. doIt(1,2);

5. }

6. // insert code here

7. }

Which three, inserted independently at line 6, will compile? (Choose three.)

A. static void doIt(int... x) { }

B. static void doIt(int... x, int y) { }

C. static void doIt(int... x, int... y) { }

D. static void doIt(int... x) { } static void doIt(int x) { }

E. static void doIt(int x) { } static void doIt(int x, int y) { }



32.Given:

1. class SuperFoo {

2. SuperFoo doStuff(int x) {

3. return new SuperFoo();

4. }

5. }

6.

7. class Foo extends SuperFoo {

8. // insert code here

9. }

Which three, inserted independently at line 8, will compile? (Choose three.)

A. int doStuff() { return 42; }

B. int doStuff(int x) { return 42; }

C. Foo doStuff(int x) { return new Foo(); }

D. Object doStuff(int x) { return new Object(); } E. SuperFoo doStuff(int x) { return new Foo(); }



33.Given:

1. class FWD {

2. int doMud(int x) { return 1; }

3. }

4. class Subaru extends FWD {

5. int doMud(int... y) { return 2; }

6. int doMud(long z) { return 3; }

7. }

8. class Race {

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

10. FWD f = new Subaru();

11. System.out.println(f.doMud(7));

12. }

13. }

What is the result?

A. 1

B. 2

C. 3

D. 7

E. Compilation fails

F. The output is NOT predictable

34.Given:

6. class Tack {

7. static short s = 17;

8. public Tack(short ss) {

9. new Tack();

10. s \*= ss;

11. }

12. public Tack() { ; }

13. }

14. public class Bridle extends Tack {

15. public Bridle(int s) { System.out.println(s + 1); } 16. public static void main(String[] args) {

17. Bridle b = new Bridle(3);

18. }

19. }

What is the result?

A. 3

B. 4

C. 31

D. 51

E. 52

F. Compilation fails

G. The stack overflows at runtime



35.Given:

1. class Test4 {

2. public static void main(String [] args) { 3. boolean x = true;

4. boolean y = false;

5. short z = 42;

6.

7. if((z++ == 42) && (y = true)) z++; 8. if((x = false) || (++z == 45)) z++; 9.

10. System.out.println("z = " + z);

11. }

12. }

What is the result?

A. z = 42

B. z = 44

C. z = 45

D. z = 46

E. Compilation fails

F. An exception is thrown at runtime

36.Given:

23. int x = 7;

24. switch (x) {

25. case 8: System.out.print("8");

26. case 7: System.out.print("7");

27. case 6: System.out.print("6");

28. default: System.out.print("def"); 29. case 9: System.out.print("9");

30. }

What is the result?

A. 7

B. 789

C. 76def

D. 76def9

E. Compilation fails



37.Given:

4. class Breakfast {

5. public static void main(String[] args) { 6. free:

7. for(int j = 3; j > 0; j--) {

8. for(int k = 0; k < 4; k++) {

9. if(j == 1) break free;

10. if(k == 2) break;

11. System.out.print(" " + j + k); 12. }

13. }

14. }

15. }

What is the result?

A. 30 31 20 21

B. 30 31 32 20 21 22

C. 30 31 32 33 20 21 22 23

D. 30 31 32 20 21 22 10 11 12

E. 30 31 32 33 20 21 22 23 10 11 12 13 F. Compilation fails



38.Given:

1. class Parser extends Utils {

2. public static void main(String [] args) { 3. try { System.out.print(new Parser().getInt("42")); 4. } catch (Exception e) {

5. System.out.println("Exc"); }

6. }

7. int getInt(String arg) {

8. return Integer.parseInt(arg);

9. }

10. }

11. class Utils {

12. int getInt(String arg) throws Exception { return 42; } 13. }

What is the result?

A. 42

B. Exc

C. 42Exc

D. Compilation fails

E. An exception is thrown at runtime



39.Given:

1. class Flow {

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

4. System.out.print("before ");

5. doRiskyThing();

6. System.out.print("after ");

7. } catch (Exception fe) {

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

9. }

10. System.out.println("done ");

11. }

12. public static void doRiskyThing() throws Exception { 13. // this code returns unless it throws an Exception 14. } }

Which two results are possible? (Choose two.)

A. before

B. before catch

C. before after done

D. before catch done

E. before after catch

F. before after catch done



40.Given:

1. class Birds {

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

3. try {

4. throw new Exception();

5. } catch (Exception e) {

6. try {

7. throw new Exception();

8. } catch (Exception e2) { System.out.print("inner "); }

9. System.out.print("middle ");

10. }

11. System.out.print("outer ");

12. }

13. }

What is the result?

A. inner

B. inner outer

C. middle outer

D. inner middle outer

E. middle inner outer

F. Compilation fails

G. An exception is thrown at runtime 

41.Given:

1. class Adder {

2. static Short s1,s2;

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

4. int x;

5. s1 = 4;

6. x = s1 + s2;

7. System.out.print(x);

8. }

9. }

What is the result?

A. 4

B. Compilation fails

C. A java.lang.ClassCastException is thrown

D. A java.lang.NullPointerException is thrown E. A java.lang.IllegalStateException is thrown

42.Given:

1. class Calc {

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

3. try {

4. int x = Integer.parseInt("42a");

5. // insert code here

6. System.out.print("oops ");

7. }

8. }

9. }

Which two, inserted independently at line 5, cause the output to be "oops "? (Choose two.)

A. } catch (ClassCastException c) {

B. } catch (IllegalStateException c) {

C. } catch (NumberFormatException n) {

D. } catch (IllegalArgumentException e) {

E. } catch (ExceptionInInitializerError e) {



43.Given:

1. class WideLoad {

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

3. float f = 3.14f;

4. new WideLoad().doIt(f);

5. }

6. void doIt(Float f) {

7. System.out.println("Float");

8. }

9. void doIt(double d) {

10. System.out.println("double");

11. }

12. }

What is the result?

A. Float

B. double

C. Compilation fails

D. The output is not predictable

E. An exception is thrown at runtime



44.Given:

5. public class Ali {

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

7. // insert code here

8. jab(y);

9. }

10. static void jab(Long x) {

11. System.out.println("Long");

12. }

13. }

And the following four code fragments:

I. int y = 7;

II. long y = 7;

III. Long y = 7;

IV. Integer y = 7;

Which code fragments, inserted independently at line 7, will compile?

A. Only fragment II

B. Only fragment III

C. Both fragments II and III

D. Fragments I, II, and III

E. Fragments II, III, and IV

F. All four fragments will compile



45.Given:

4. class Slug {

5. static void crawl() { System.out.print("crawling "); } 6. }

7. public class BananaSlug extends Slug {

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

9. Slug[] sa = { new Slug(), new BananaSlug() }; 10. for(Slug s: sa)

11. crawl();

12. }

13. static void crawl() { System.out.print("shuffling "); } 14. }

What is the result?

A. crawling crawling

B. crawling shuffling

C. shuffling shuffling

D. Compilation fails

E. An exception is thrown at runtime

46.Given:

1. class Animal { Animal getOne() { return new Animal(); } } 2. class Dog extends Animal {

3. // insert code here

4. }

5.

6. class AnimalTest {

7. public static void main(String [] args) { 8. Animal [] animal = { new Animal(), new Dog() } ; 9. for( Animal a : animal) {

10. Animal x = a.getOne();

11. }

12. }

13. }

And the code:

3a. Dog getOne() { return new Dog(); }

3b. Animal getOne() { return new Dog(); }

Which, inserted at line 3, will compile and run with no exceptions?

A. Only line 3a

B. Only line 3b

C. Either line 3a or 3b

D. Both line 3a and 3b added together



47.Given:

4. public class Utility {

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

6. System.out.print(x + " ");

7. new Utility().go(x);

8. System.out.println(x);

9. }

10. private void go(int x) {

11. x += 5;

12. }

13. int x = 7;

14. }

What is the result?

A. 7 5

B. 7 7

C. 7 12

D. Compilation fails

E. An exception is thrown at runtime



48.Given:

6. class Bird {

7. protected void talk() { System.out.print("chirp "); } 8. }

9. public class Macaw extends Bird {

10. public static void main(String [] args) { 11. Bird [] birds = {new Bird(), new Macaw()}; 12. for( Bird b : birds)

13. b.talk();

14. }

15. void talk() { System.out.print("hello "); } 16. }

What is the result?

A. chirp chirp

B. chirp hello

C. hello hello

D. hello chirp

E. Compilation fails

F. An exception is thrown at runtime



49.Given:

4. class Gadget {

5. Gadget openStuff() { s = s + "g "; return this; } 6. static String s = "";

7. }

8. public class Opener extends Gadget {

9. public static void main(String[] args) { 10. Gadget g1 = new Gadget();

11. Gadget g2 = new Opener();

12. g1.openStuff();

13. g2.openStuff();

14. System.out.println(s);

15. }

16. Opener openStuff() { s = s + "o "; return this; } 17. }

What is the result?

A. g g

B. g o

C. o g

D. o o

E. Compilation fails

F. An exception is thrown at runtime



50.Given:

2. class Cat {

3. Cat(int c) { System.out.print("cat" + c + " "); }

4. }

5. class SubCat extends Cat {

6. SubCat(int c) { super(5); System.out.print("cable "); } 7. SubCat() { this(4); }

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

9. SubCat s = new SubCat();

10. }

11. }

What is the result?

A. cat5

B. cable

C. cable cat5

D. cat5 cable

E. Compilation fails

F. An exception is thrown at runtime



51.A programmer wants to develop an application in which Fizzlers are a kind of Whoosh, and Fizzlers also fulfill the contract of Oompahs. In addition, Whooshes are composed with several Wingits.

Which code represents this design?

A. class Wingit { }

class Fizzler extends Oompah implements Whoosh { }

interface Whoosh {

Wingits [] w;

}

class Oompah { }

B. class Wingit { }

class Fizzler extends Whoosh implements Oompah { }

class Whoosh {

Wingits [] w;

}

interface Oompah { }

C. class Fizzler { }

class Wingit extends Fizzler implements Oompah { }

class Whoosh {

Wingits [] w;

}

interface Oompah { }

D. interface Wingit { }

class Fizzler extends Whoosh implements Wingit { }

class Wingit {

Whoosh [] w;

}

class Whoosh { }



52.Given:

2. import java.util.\*;

3. class MyInfo implements Comparable {

4. MyInfo(String s) { desc = s; }

5. String desc;

6. // insert code here

7. public String toString() { return desc; }

8. }

Which method, inserted at line 6, correctly implements java.util.Comparable?

A. int compare(MyInfo m) {

return desc.compareTo(m.desc); }

B. int compareTo(MyInfo m) {

return desc.compareTo(m.desc); }

C. int compareable(MyInfo m) {

return desc.compareTo(m.desc); }

D. int compare(Object o) {

return desc.compareTo(o.toString()); }

E. int compareTo(Object o) {

return desc.compareTo(o.toString()); }

F. int compareable(Object o) {

return desc.compareTo(o.toString()); }



53.Given:

3. public class Doctor {

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

5. String s = "";

6. int x = 2;

7. if((7 > (4\*2)) ^ (5 != 4)) s += "1st ";

8. if((5 < 7) ^ (7 > 5)) s += "2nd ";

9. if(((4 \* x++) < 9) ^ (x > 3)) s += "3rd ";

10. System.out.println(s);

11. }

12. }

What is the result?

A. 1st

B. 2nd

C. 3rd

D. 1st 2nd

E. 1st 3rd

F. 2nd 3rd

G. No output is produced



54.Given:

4. import java.util.\*;

5. class Looking {

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

7. String[] sa = {"d", "c", "a", "b" };

8. int x = Arrays.binarySearch(sa, "b");

9. Arrays.sort(sa);

10. int y = Arrays.binarySearch(sa, "b");

11. System.out.println(x + " " + y);

12. }

13. }

Which two results are possible? (Choose two.)

A. 7 0

B. 7 1

C. 7 3

D. -1 0

E. -1 1

F. -1 3

55.Given two files:

1. package x;

2. public class X {

3. // insert code here

4. }

7. package x;

8. public class Find4 {

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

10. X myX = new X();

11. myX.doX();

12. }

13. }

And the four methods:

1. public static void doX() { System.out.print("doX 1 "); }
2. static void doX() { System.out.print("doX 2 "); }
3. protected static void doX() { System.out.print("doX 3 "); }
4. private static void doX() { System.out.print("doX 4 "); }

Which methods, inserted independently at line 3, will compile?

A. Only method I

B. Only methods I and II

C. Only methods I and III

D. Only methods II and III

E. Only methods I, II, and III

F. All the methods will compile

56.Given:

1. import java.util.\*;

2. class Pow {

3. static String [] wow = {"Bamm", "Biff"}; 4. public static void main(String [] yikes) { 5. if(Arrays.equals(yikes,wow))

6. System.out.print("got a match? "); 7. if(yikes == wow)

8. System.out.println("sure chief"); 9. }

10. }

And the command line:

java Pow Bamm Biff

What is the result?

A. got a match?

B. Compilation fails

C. No output is produced

D. got a match? sure chief

E. An exception is thrown at runtime 

57.Given:

1. class Flibitz {

2. public static void main(String [] args) { 3. int grop = 7;

4. new Flibitz().go(grop);

5. System.out.print(grop);

6. }

7. void go(int grop) {

8. if(++grop > 7) grop++;

9. System.out.print(grop);

10. }

11. }

What is the result?

A. 77

B. 79

C. 97

D. 99

E. Compilation fails

F. An exception is thrown at runtime

58. How many of these lines fail to compile?

Predicate<String> pred1 = s -> false;

Predicate<String> pred2 = (s) -> false;

Predicate<String> pred3 = String s -> false;

Predicate<String> pred4 = (String s) -> false;

A. One

B. Two

C. Three

D. Four

59. What does the following do?

public class Shoot {

interface Target {

boolean needToAim(double angle);

}

static void prepare(double angle, Target t) { boolean ready = t.needToAim(angle); // k1

System.out.println(ready);

}

public static void main(String[] args) {

prepare(45, d -> d > 5 || d < -5); // k2

}

}

A. It prints true.

B. It prints false.

C. It doesn’t compile due to line k1.

D. It doesn’t compile due to line k2.

60. Which of the following can fill in the blank to make the code compile?

import java.util.function.\*;

public class Card {

public static void main(String[] s) {

Predicate<String> pred = -> true;

}

}

A. (Integer i)

B. (Object o)

C. (String s)

D. None of the above

61. What is the output of the following?

LocalDate xmas = LocalDate.of(2016, 12, 25);

xmas.plusDays(-1);

System.out.println(xmas.getDayOfMonth());

A. 24

B. 25

C. 26

D. None of the above

62. Which of these periods represents a larger amount of time?

Period period1 = Period.ofWeeks(1).ofDays(3);

Period period2 = Period.ofDays(10);

A. period1

B. period2

C. They represent the same length of time.

D. None of the above. This code does not compile.

63. What is the result of the following?

import java.time.\*;

import java.time.format.\*;

public class HowLong {

public static void main(String[] args) {

LocalDate newYears = LocalDate.of(2017, 1, 1);

Period period = Period.ofDays(1);

DateTimeFormatter format= DateTimeFormatter.ofPattern("MM-dd-yyyy"); System.out.print(format.format(newYears.minus(period)));

}

}

A. 01-01-2017

B. 12-31-2016

C. The code does not compile.

D. The code compiles but throws an exception at runtime.

64. How many lines does this code output?

import java.util.\*;

import java.util.function.\*;

public class PrintNegative {

public static void main(String[] args) {

List<String> list = new ArrayList<>();

list.add("-5");

list.add("0");

list.add("5");

print(list, e -> e < 0);

}

public static void print(List<String> list, Predicate<Integer> p) { for (String num : list)

if (p.test(num))

System.out.println(num);

}

}

A. One

B. Two

C. None. The code does not compile.

D. None. The code throws an exception at runtime.

65. What is the result of the following?

LocalDate xmas = LocalDate.of(2016, 12, 25);

xmas.setYear(2017);

System.out.println(xmas.getYear());

A. 2016

B. 2017

C. The code does not compile.

D. The code compiles but throws an exception at runtime.

66. Given:

11. class Snoochy {

12. Boochy booch;

13. public Snoochy() { booch = new Boochy(this); }

14.}

15. class Boochy {

16. Snoochy snooch;

17. public Boochy(Snoochy s) { snooch = s;}

18.}

And the statements:

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

22. Snoochy snoog = new Snoochy();

23. snoog = null;

24. // more code here

25. }

Which statement is true about the objects referenced by snoog, snooch and booch immediately after line 23 executes?

A - None of these objects are eligible for garbage collection.

B - Only the object referenced by booch is eligible for garbage collection.

C - Only the object referenced by snoog is eligible for garbage collection.

D - Only the object referenced by snooch is eligible for garbage collection.

E - The objects referenced by snooch and booch are eligible for garbage collection.

67. What is the result?

11. class Person {

12. String name = "No name";

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

14.}

15. class Employee extends Person {

16. String empID = "0000";

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

18.}

19. public class EmployeeTest {

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

21. Employee e = new Employee("4321");

22. System.out.println(e.empID);

23. }

24.}

A - 4321

B - 0000

C - An exception is thrown at runtime.

D - Compilation fails because of an error in line 17

68.Given a class Repetition and another class Demo:

1. package utils;

2. public class Repetition {

3. public static String twice(String s) {return s + s;}

4.}

1. // insert code here

2. public class Demo {

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

4. System .out.println(twice("pizza"));

5. }

6. }

Which code should be inserted at line 1 of Demo.java to compile and run Demo to print "pizzapizza"?

A - import utils.\*;

B - static import utils.\*;

C - import utils.Repetition.\*;

D - static import utils.Repetition.\*;

E - import utils.Repetition.twice();

F - import static utils.Repetition.twice;

G - static import utils.Repetition.twice;

69.Given:

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

12. try{

13. args = null;

14. args[0] = "test";

15. System.out.println(args[0]);

16. } catch (Exception ex) {

17. System.out.println("Exception");

18. } catch (NullPointerException npe) {

19. System.out.println("NullPointerException");

20. }

21. }

What is the result?

A - test

B - Exception

C - Compilation fails.

D - NullPointerException

70. Given:

12. public class Wow {

13. public static void go(short n) {System.out.println("short");} 14. public static void go(Short n) {System.out.println("SHORT");} 15. public static void go(Long n) {System.out.println(" LONG");} 16. public static void main(String [ ] args) {

17. Short y = 6;

18. int z = 7;

19. go(y);

20. go(z);

21. }

22.}

What is the result?

A - short LONG

B - SHORT LONG

C - Compilation fails.

D - An exception is thrown at runtime.