**01. Given:**

8. public class test {

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

10. assert a.length == 1;

11. }

12. }

Which two will produce an AssertionError? (Choose two.)

A. java -ea test file1 file2

B. java -ea test

C. java test

D. java test file1

E. java -ea test file1

F. java -ea:test test file1

Answer: A,B

**02. Given:**

12. String csv = "Sue,5,true,3";

13. Scanner scanner = new Scanner( csv );

14. scanner.useDelimiter(",");

15. int age = scanner.nextInt();

What is the result?

A. After line 15, the value of age is 3.

B. Compilation fails.

C. An exception is thrown at runtime.

D. After line 15, the value of age is 5.

Answer: C

**03. Given:**

35. String #name = "Jane Doe";

36. int $age = 24;

37. Double \_height = 123.5;

38. double ~temp = 37.5;

Which two statements are true? (Choose two.)

A. Line 35 will not compile.

B. Line 38 will not compile.

C. Line 37 will not compile.

D. Line 36 will not compile.

Answer: A,B

**04. Given:**

10. class Nav{

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

12. }

13. public class Sprite{

14. // insert code here

15. }

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

A. Nav.Direction d = Nav.Direction.NORTH;

B. Nav.Direction d = NORTH;

C. Direction d = NORTH;

D. Direction d = Direction.NORTH;

Answer: A

**05. 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. A NumberFormatException occurs at runtime.

B. An IllegalStateException occurs at runtime.

C. Compilation fails because of an error in line 9.

D. A NullPointerException occurs at runtime.

E. Compilation fails because of an error in line 5.

F. The value "4" is printed at the command line.

Answer: D

**06. 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 Example values cannot be used in a raw java.util.HashMap; instead, the programmer must

use a java.util.EnumMap.

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

guaranteed to be less than one.

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.

Answer: A

**07. Given:**

11. public enum Title {

12. MR("Mr."), MRS("Mrs."), MS("Ms.");

13. private final String title;

14. private Title(String t) { title = t; }

15. public String format(String last, String first) {

16. return title + " " + first + " " + last;

17. }

18. }

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

20. System.out.println(Title.MR.format("Doe", "John"));

21. }

What is the result?

A. An exception is thrown at runtime.

B. Compilation fails because of an error in line 15.

C. Compilation fails because of an error in line 12.

D. Compilation fails because of an error in line 20.

E. Mr. John Doe

Answer: E

**08. Given:**

11. String test = "This is a test";

12. String[] tokens = test.split("\s");

13. System.out.println(tokens.length);

What is the result?

A. 4

B. Compilation fails.

C. 0

D. An exception is thrown at runtime.

E. 1

Answer: B

**09. 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 in the test package

B. any class

C. only the Target class

D. any class that extends Target

Answer: A

**10. Given**

1. public class TestString3 { 2. public static void main(String[] args) { 3. // insert code here

5. System.out.println(s); 6. } 7. } Which two code fragments, inserted independently at line 3,

generate the output 4247? (Choose two.)

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

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

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

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

C. String s = "123456789";

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

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

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

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

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

Answer: A,B

**11. Given**

11. String[] elements = { "for", "tea", "too" };

12. String first = (elements.length > 0) ? elements[0] : null;

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The variable first is set to elements[0].

D. The variable first is set to null.

Answer: C

**12. Given:**

55. int [] x = {1, 2, 3, 4, 5};

56. int y[] = x;

57. System.out.println(y[2]);

Which statement is true?

A. Compilation will fail because of an error in line 55.

B. Line 57 will print the value 2.

C. Compilation will fail because of an error in line 56.

D. Line 57 will print the value 3.

Answer: D

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

prevent duplicates.

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

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

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

hashCode method.

Answer: C

**14. Given:**

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

12. Integer i = new Integer(1) + new Integer(2);

13. switch(i) {

14. case 3: System.out.println("three"); break;

15. default: System.out.println("other"); break;

16. }

17. }

What is the result?

A. three

B. Compilation fails because of an error on line 12.

C. Compilation fails because of an error on line 13.

D. other

E. An exception is thrown at runtime.

F. Compilation fails because of an error on line 15.

Answer: A

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

A. class Man extends Dog { }

B. class Man { private Dog<bestFriend>; }

C. class Man implements Dog { }

D. class Man { private Dog bestFriend; }

E. class Man { private BestFriend<dog>; }

F. class Man { private BestFriend dog; }

Answer: D

**16. Given**

20. public class CreditCard {

21.

22. private String cardID;

23. private Integer limit;

24. public String ownerName;

25.

26. public void setCardInformation(String cardID,

27. String ownerName,

28. Integer limit) {

29. this.cardID = cardID;

30. this.ownerName = ownerName;

31. this.limit = limit;

32. }

33. }

Which statement is true?

A. The cardID and limit variables break polymorphism.

B. The code demonstrates polymorphism.

C. The ownerName variable breaks encapsulation.

D. The setCardInformation method breaks encapsulation.

E. The class is fully encapsulated.

Answer: C

**17. Given**

10. class Foo {

11. static void alpha() { /\* more code here \*/ }

12. void beta() { /\* more code here \*/ }

13. }

Which two statements are true? (Choose two.)

A. Foo.beta() is a valid invocation of beta().

B. Method alpha() can directly call method beta().

C. Method beta() can directly call method alpha().

D. Foo.alpha() is a valid invocation of alpha().

Answer: C,D

18. Given the command line java Pass2 and:

15. public class Pass2 {

16. public void main(String [] args) {

17. int x = 6;

18. Pass2 p = new Pass2();

19. p.doStuff(x);

20. System.out.print(" main x = " + x);

21. }

22.

23. void doStuff(int x) {

24. System.out.print(" doStuff x = " + x++);

25. }

26. }

What is the result?

A. doStuff x = 6 main x = 6

B. doStuff x = 6 main x = 7

C. doStuff x = 7 main x = 7

D. doStuff x = 7 main x = 6

E. Compilation fails.

F. An exception is thrown at runtime.

Answer: A

**19. Given:**

11. public class Test {

12. public enum Dogs {collie, harrier, shepherd};

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

14. Dogs myDog = Dogs.shepherd;

15. switch (myDog) {

16. case collie:

17. System.out.print("collie ");

18. case default:

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

20. case harrier:

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

22. }

23. }

24. }

What is the result?

A. shepherd

B. An exception is thrown at runtime.

C. retriever

D. retriever harrier

E. Compilation fails.

F. harrier

Answer: E

**20. A programmer needs to create a logging method that can accept an arbitrary number of**

arguments. For example, it may be called in these ways:

logIt("log message1");

logIt("log message2","log message3");

logIt("log message4","log message5","log message6");

Which declaration satisfies this requirement?

A. public void logIt(String \* msgs)

B. public void logIt(String... msgs)

C. public void logIt(String [] msgs)

D. public void logIt(String msg1, String msg2, String msg3)

Answer: B

**21. Given:**

11. public class Person {

12. private name;

13. public Person(String name) {

14. this.name = name;

15. }

16. public int hashCode() {

17. return 420;

18. }

19. }

Which statement is true?

A. Deleting a Person key from a HashMap will delete all map entries for all keys of type Person.

B. The time to find the value from HashMap with a Person key depends on the size of the map.

C. The time to determine whether a Person object is contained in a HashSet is constant and does

NOT depend on the size of the map.

D. Inserting a second Person object into a HashSet will cause the first Person object to be

removed as a duplicate.

Answer: B

**22. Given:**

10. class Line {

11. public class Point { public int x,y;}

12. public Point getPoint() { return new Point(); }

13. }

14. class Triangle {

15. public Triangle() {

16. // insert code here

17. }

18. }

Which code, inserted at line 16, correctly retrieves a local instance of a Point object?

A. Line.Point p = Line.getPoint();

B. Point p = (new Line()).getPoint();

C. Point p = Line.getPoint();

D. Line.Point p = (new Line()).getPoint();

Answer: D

**23. 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. only the Target class

B. any class in the test package

C. any class that extends Target

D. any class

Answer: B

**24. 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. Compilation fails.

C. SHORT LONG

D. An exception is thrown at runtime.

Answer: B

**25. Given:**

11. public class Ball{

12. public enum Color { RED, GREEN, BLUE };

13. public void foo(){

14. // insert code here

15. { System.out.println(c); }

16. }

17. }

Which code inserted at line 14 causes the foo method to print RED, GREEN, and BLUE?

A. for( Color c = Color[0]; c <= Color[2]; c++ )

B. for( Color c ; c.hasNext() ; c.next() )

C. for( Color c = Color.RED; c <= Color.BLUE; c++ )

D. for( Color c = RED; c <= BLUE; c++ )

E. for( Color c : Color.values() )

Answer: E

**26. Given:**

15. public class Yippee {

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

17. for(int x = 1; x < args.length; x++) {

18. System.out.print(args[x] + " ");

19. }

20. }

21. }

and two separate command line invocations:

java Yippee

java Yippee 1 2 3 4

What is the result?

A. An exception is thrown at runtime.

2 3 4

B. No output is produced.

2 3 4

C. No output is produced.

1 2 3

D. An exception is thrown at runtime.

1 2 3 4

E. An exception is thrown at runtime.

1 2 3

F. No output is produced.

1 2 3 4

Answer: B

**27. Given:**

10. public class Bar {

11. static void foo( int... x ) {

12. // insert code here

13. }

14. }

Which two code fragments, inserted independently at line 12, will allow the class to compile?

(Choose two.)

A. for( int i=0; i< x.length; i++ ) System.out.println(x[i]);

B. for( int z : x ) System.out.println(z);

C. foreach( x ) System.out.println(z);

D. while( x.hasNext() ) System.out.println( x.next() );

Answer: A,B

**28. Given:**

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

12. String str = "null";

13. if (str == null) {

14. System.out.println("null");

15. } else (str.length() == 0) {

16. System.out.println("zero");

17. } else {

18. System.out.println("some");

19. }

20. }

What is the result?

A. An exception is thrown at runtime.

B. null

C. Compilation fails.

D. zero

E. some

Answer: C

**29. Given:**

12. public class Yippee2 {

13.

14. static public void main(String [] yahoo) {

15. for(int x = 1; x < yahoo.length; x++) {

16. System.out.print(yahoo[x] + " ");

17. }

18. }

19. }

and the command line invocation:

java Yippee2 a b c

What is the result?

A. a b c

B. Compilation fails.

C. a b

D. An exception is thrown at runtime.

E. b c

Answer: E

**15. Given:**

11. public static void test(String str) {

12. int check = 4;

13. if (check = str.length()) {

14. System.out.print(str.charAt(check -= 1) +", ");

15. } else {

16. System.out.print(str.charAt(0) + ", ");

17. }

18. }

and the invocation:

21. test("four");

22. test("tee");

23. test("to");

What is the result?

A. An exception is thrown at runtime.

B. r, e, o,

C. Compilation fails.

D. r, t, t,

Answer: C

**30. Given:**

11. String test = "This is a test";

12. String[] tokens = test.split("\s");

13. System.out.println(tokens.length);

What is the result?

A. An exception is thrown at runtime.

B. 1

C. 4

D. Compilation fails.

E. 0

Answer: D

**31. Given:**

11. Float pi = new Float(3.14f);

12. if (pi > 3) {

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

14. }

15. else {

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

17. }

18. finally {

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

20. }

What is the result?

A. pi is not bigger than 3. Have a nice day.

B. pi is bigger than 3. Have a nice day.

C. pi is bigger than 3.

D. Compilation fails.

E. An exception occurs at runtime.

Answer: D

**32. Given:**

11. Float pi = new Float(3.14f);

12. if (pi > 3) {

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

14. }

15. else {

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

17. }

18. finally {

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

20. }

What is the result?

A. Compilation fails.

B. An exception occurs at runtime.

C. pi is not bigger than 3. Have a nice day.

D. pi is bigger than 3. Have a nice day.

E. pi is bigger than 3.

Answer: A

**33. 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. 1 2

B. An exception is thrown at runtime.

C. Compilation fails.

D. 2

E. 1 2 3

F. 3

G. 2 3

Answer: G

34. Given:

11. public static void test(String str) {

12. if (str == null | str.length() == 0) {

13. System.out.println("String is empty");

14. } else {

15. System.out.println("String is not empty");

16. }

17. }

And the invocation:

31. test(null);

What is the result?

A. "String is not empty" is printed to output.

B. Compilation fails because of an error in line 12.

C. An exception is thrown at runtime.

D. "String is empty" is printed to output.

Answer: C

38. } while (x < 10);

How many times will line 37 be executed?

A. more than ten times

B. ten times

C. zero times

D. one to nine times

Answer: A

35. Given:

12. public class Test {

13. public enum Dogs {collie, harrier};

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

15. Dogs myDog = Dogs.collie;

16. switch (myDog) {

17. case collie:

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

19. case harrier:

20. System.out.print("harrier ");

21. }

22. }

23. }

What is the result?

A. Compilation fails.

B. collie

C. An exception is thrown at runtime.

D. harrier

E. collie harrier

Answer: E

36. Given:

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

12. for (int i = 0; i <= 10; i++) {

13. if (i > 6) break;

14. }

15. System.out.println(i);

16. }

What is the result?

A. Compilation fails.

B. 10

C. 7

D. 6

E. 11

F. An exception is thrown at runtime.

Answer: A

37. Given:

11. public static Iterator reverse(List list) {

12. Collections.reverse(list);

13. return list.iterator();

14. }

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

16. List list = new ArrayList();

17. list.add("1"); list.add("2"); list.add("3");

18. for (Object obj: reverse(list))

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

20. }

What is the result?

A. The code runs with no output.

B. 3, 2, 1,

C. An exception is thrown at runtime.

D. Compilation fails.

E. 1, 2, 3,

Answer: D

37. Given:

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

12. String str = "null";

13. if (str == null) {

14. System.out.println("null");

15. } else (str.length() == 0) {

16. System.out.println("zero");

17. } else {

18. System.out.println("some");

19. }

20. }

What is the result?

A. An exception is thrown at runtime.

B. some

C. zero

D. Compilation fails.

E. null

Answer: D

38. Given:

11. public class Person {

12. private String name;

13. public Person(String name) {

14. this.name = name;

15. }

16. public boolean equals(Object o) {

17. if ( ! o instanceof Person ) return false;

18. Person p = (Person) o;

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

20. }

21. }

Which statement is true?

A. A HashSet could contain multiple Person objects with the same name.

B. If a HashSet contains more than one Person object with name="Fred", then removing another

Person, also with name="Fred", will remove them all.

C. All Person objects will have the same hash code because the hashCode method is not

overridden.

D. Compilation fails because the hashCode method is not overridden.

Answer: A

39. Given:

15. public class Yippee {

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

17. for(int x = 1; x < args.length; x++) {

18. System.out.print(args[x] + " ");

19. }

20. }

21. } a

nd two separate command line invocations:

java Yippee

java Yippee 1 2 3 4

What is the result?

A. An exception is thrown at runtime.

1 2 3 4

B. No output is produced.

1 2 3

C. No output is produced.

2 3 4

D. No output is produced.

1 2 3 4

E. An exception is thrown at runtime.

1 2 3

F. An exception is thrown at runtime.

2 3 4

Answer: C

40. Given:

10. public class Bar {

11. static void foo( int... x ) {

12. // insert code here

13. }

14. }

Which two code fragments, inserted independently at line 12, will allow the class to compile?

(Choose two.)

A. while( x.hasNext() ) System.out.println( x.next() );

B. for( int i=0; i< x.length; i++ ) System.out.println(x[i]);

C. foreach( x ) System.out.println(z);

D. for( int z : x ) System.out.println(z);

Answer: B,D

41. Given:

25. int x = 12;

26. while (x < 10) {

27. x--;

28. }

29. System.out.print(x);

What is the result?

A. Line 29 will never be reached.

B. 12

C. 0

D. 10

Answer: B

42. Given:

25. int x = 12;

26. while (x < 10) {

27. x--;

28. }

29. System.out.print(x);

What is the result?

A. 12

B. 0

C. Line 29 will never be reached.

D. 10

Answer: A

43. Given:

10. int x = 0;

11. int y = 10;

12. do {

13. y--;

14. ++x;

15. } while (x < 5);

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

What is the result?

A. 6,6

B. 5,5

C. 6,5

D. 5,6

Answer: B

44. Given:

23. int z = 5;

24.

25. public void stuff1(int x) {

26. assert (x > 0);

27. switch(x) {

28. case 2: x = 3;

29. default: assert false; } }

30.

31. private void stuff2(int y) { assert (y < 0); }

32.

33. private void stuff3() { assert (stuff4()); }

34.

35. private boolean stuff4() { z = 6; return false; }

Which statement is true?

A. Only the assert statement on line 31 is used appropriately.

B. The assert statements on lines 26, 29, and 31 are used appropriately.

C. The assert statements on lines 29 and 31 are used appropriately.

D. The assert statements on lines 29 and 33 are used appropriately.

E. The assert statements on lines 26 and 29 are used appropriately.

F. All of the assert statements are used appropriately.

G. The assert statements on lines 29, 31, and 33 are used appropriately.

Answer: C

45. Given:

11. String test = "a1b2c3";

12. String[] tokens = test.split("\\d");

13. for(String s: tokens) System.out.print(s + " ");

What is the result?

A. a1 b2 c3

B. The code runs with no output.

C. a b c

D. An exception is thrown at runtime.

E. 1 2 3

F. a1b2c3

G. Compilation fails.

Answer: C

46. Which two code fragments correctly create and initialize a static array of int elements? (Choose

two.)

A. static final int[] a = { 100,200 };

B. static final int[] a;

static { a=new int[2]; a[0]=100; a[1]=200; }

C. static final int[] a;

static void init() { a = new int[3]; a[0]=100; a[1]=200; }

D. static final int[] a = new int[2]{ 100,200 };

Answer: A,B

47. Given:

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

12. Object obj = new int[] { 1, 2, 3 };

13. int[] someArray = (int[])obj;

14. for (int i : someArray) System.out.print(i + " ");

15. }

What is the result?

A. Compilation fails because of an error in line 13.

B. A ClassCastException is thrown at runtime.

C. 1 2 3

D. Compilation fails because of an error in line 14.

E. Compilation fails because of an error in line 12.

Answer: C

48. Given this method in a class:

21. public String toString() {

22. StringBuffer buffer = new StringBuffer();

23. buffer.append('<');

24. buffer.append(this.name);

25. buffer.append('>');

26. return buffer.toString();

27. }

Which statement is true?

A. The programmer can replace StringBuffer with StringBuilder with no other changes.

B. This code will perform poorly. For better performance, the code should be rewritten:

return "<" + this.name + ">";

C. This code is NOT thread-safe.

D. This code will perform well and converting the code to use StringBuilder will not enhance the

performance.

Answer: A

49. Given:

1. public class Blip {

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

3. }

4. class Vert extends Blip {

5. // insert code here

6. }

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

A. protected int blipvert(long x) { return 0; }

B. protected long blipvert(int x) { return 0; }

C. private int blipvert(long x) { return 0; }

D. private int blipvert(int x) { return 0; }

E. public int blipvert(int x) { return 0; }

F. protected long blipvert(long x) { return 0; }

G. protected long blipvert(int x, int y) { return 0; }

Answer: A,C,E,F,G

50. Given:

12. System.out.format("Pi is approximately %d.", Math.PI);

What is the result?

A. An exception is thrown at runtime.

B. Compilation fails.

C. Pi is approximately 3.

D. Pi is approximately 3.141593.

Answer: A

51. Given this method in a class:

21. public String toString() {

22. StringBuffer buffer = new StringBuffer();

23. buffer.append('<');

24. buffer.append(this.name);

25. buffer.append('>');

26. return buffer.toString();

27. }

Which statement is true?

A. This code will perform poorly. For better performance, the code should be rewritten:

return "<" + this.name + ">";

B. This code is NOT thread-safe.

C. The programmer can replace StringBuffer with StringBuilder with no other changes.

D. This code will perform well and converting the code to use StringBuilder will not enhance the

performance.

Answer: C

52. Given:

1. public class Blip {

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

3. }

4. class Vert extends Blip {

5. // insert code here

6. }

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

A. private int blipvert(long x) { return 0; }

B. protected long blipvert(int x, int y) { return 0; }

C. protected long blipvert(long x) { return 0; }

D. protected long blipvert(int x) { return 0; }

E. public int blipvert(int x) { return 0; }

F. private int blipvert(int x) { return 0; }

G. protected int blipvert(long x) { return 0; }

Answer: A,B,C,E,G

53. Given:

10. class One {

11. public One foo() { return this; }

12. }

13. class Two extends One {

14. public One foo() { return this; }

15. }

16. class Three extends Two {

17. // insert method here

18. }

Which two methods, inserted individually, correctly complete the Three class? (Choose two.)

A. public int foo() { return 3; }

B. public Two foo() { return this; }

C. public void foo() {}

D. public One foo() { return this; }

E. public Object foo() { return this; }

Answer: B,D

54. Given:

10. class Line {

11. public static class Point {}

12. }

13.

14. class Triangle {

15. // insert code here

16. }

Which code, inserted at line 15, creates an instance of the Point class defined in Line?

A. Line l = new Line() ; l.Point p = new l.Point();

B. Line.Point p = new Line.Point();

C. The Point class cannot be instatiated at line 15.

D. Point p = new Point();

Answer: B

55. Given:

11. public class Yikes {

12.

13. public static void go(Long n) {System.out.println("Long ");}

14. public static void go(Short n) {System.out.println("Short ");}

15. public static void go(int n) {System.out.println("int ");}

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

17. short y = 6;

18. long z = 7;

19. go(y);

20. go(z);

21. }

22. }

What is the result?

A. An exception is thrown at runtime.

B. int Long

C. Compilation fails.

D. Short Long

Answer: B

56. Given:

11. public abstract class Shape {

12. private int x;

13. private int y;

14. public abstract void draw();

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

16. this.x = x;

17. this.y = y;

18. }

19. }

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

A. public class Circle extends Shape {

private int radius;

public void draw();

}

B. public abstract class Circle extends Shape {

private int radius;

}

C. public class Circle implements Shape {

private int radius;

}

D. public abstract class Circle implements Shape {

private int radius;

public void draw() { /\* code here \*/ }

E. public abstract class Circle implements Shape {

private int radius;

public void draw();

}

F. public class Circle extends Shape {

private int radius;

public void draw() {/\* code here \*/}

Answer: B,F

57. Given:

10. class One {

11. public One() { System.out.print(1); }

12. }

13. class Two extends One {

14. public Two() { System.out.print(2); }

15. }

16. class Three extends Two {

17. public Three() { System.out.print(3); }

18. }

19. public class Numbers{

20. public static void main( String[] argv ) { new Three(); }

21. }

What is the result when this code is executed?

A. 321

B. 3

C. 1

D. 123

E. The code runs with no output.

Answer: D

58. Given:

11. public abstract class Shape {

12. int x;

13. int y;

14. public abstract void draw();

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

16. this.x = x;

17. this.y = y;

18. }

19. }

and a class Circle that extends and fully implements the Shape class.

Which is correct?

A. Shape s = new Circle();

s.setAnchor(10,10);

s.draw();

B. Circle c = new Circle();

c.Shape.setAnchor(10,10);

c.Shape.draw();

C. Circle c = new Shape();

c.setAnchor(10,10);

c.draw();

D. Shape s = new Shape();

s.setAnchor(10,10);

s.draw();

E. Shape s = new Circle();

s->setAnchor(10,10);

s->draw();

Answer: A

59. Given:

10. abstract class A {

11. abstract void a1();

12. void a2() { }

13. }

14. class B extends A {

15. void a1() { }

16. void a2() { }

17. }

18. class C extends B { void c1() { } }

and:

A x = new B(); C y = new C(); A z = new C();

What are four valid examples of polymorphic method calls? (Choose four.)

A. x.a2();

B. x.a1();

C. z.a2();

D. z.a1();

E. y.c1();

F. z.c1();

Answer: A,B,C,D

60. Assuming that the serializeBanana2() and the deserializeBanana2() methods will correctly use

Java serialization and given:

13. import java.io.\*;

14. class Food {Food() { System.out.print("1"); } }

15. class Fruit extends Food implements Serializable {

16. Fruit() { System.out.print("2"); } }

17. public class Banana2 extends Fruit { int size = 42;

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

19. Banana2 b = new Banana2();

20. b.serializeBanana2(b); // assume correct serialization

21. b = b.deserializeBanana2(b); // assume correct

22. System.out.println(" restored " + b.size + " "); }

23. // more Banana2 methods

24. }

What is the result?

A. An exception is thrown at runtime.

B. 12 restored 42

C. 1212 restored 42

D. Compilation fails.

E. 121 restored 42

F. 1 restored 42

Answer: E

61. Given

11. public interface Status {

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

13. }

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

A. private

B. native

C. public

D. protected

E. abstract

F. final

G. static

Answer: C,F,G

62. 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. Dog is-a Jumper

B. Beagle has-a Jumper

C. Cat has-a Animal

D. Dog is-a Animal

E. Cat is-a Jumper

F. Beagle has-a Tail

G. Cat is-a Animal

Answer: D,E,F

63. Given:

11. interface DeclareStuff {

12. public static final int EASY = 3;

13. void doStuff(int t); }

14. public class TestDeclare implements DeclareStuff {

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

16. int x = 5;

17. new TestDeclare().doStuff(++x);

18. }

19. void doStuff(int s) {

20. s += EASY + ++s;

64. 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?

A. A ClassCastException occurs in line 31.

B. The value of all four objects prints in natural order.

C. Compilation fails due to an error in line 23.

D. A ClassCastException occurs in line 29.

E. Compilation fails due to an error in line 29.

Answer: D

65. Given

21. System.out.println("s " + s);

22. }

23. }

What is the result?

A. Compilation fails.

B. s 16

C. s 14

D. An exception is thrown at runtime.

E. s 10

Answer: A

66. Given:

10. interface A { public int getValue(); }

11. class B implements A {

12. public int getValue() { return 1; }

13. }

14. class C extends B {

15. // insert code here

16. }

Which three code fragments, inserted individually at line 15, make use of polymorphism? (Choose

three.)

A. public void add(A a, B b) { a.getValue(); }

B. public void add(C c1, C c2) { c1.getValue(); }

C. public void add(C c) { c.getValue(); }

D. public void add(B b) { b.getValue(); }

E. public void add(A a) { a.getValue(); }

Answer: A,D,E

66. Given classes defined in two different files:

1. package util;

2. public class BitUtils {

3. public static void process(byte[]) { /\* more code here \*/ }

4. }

1. package app;

2. public class SomeApp {

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

4. byte[] bytes = new byte[256];

5. // insert code here

6. }

7. }

What is required at line 5 in class SomeApp to use the process method of BitUtils?

A. process(bytes);

B. import util.BitUtils.\*; process(bytes);

C. BitUtils.process(bytes);

D. SomeApp cannot use methods in BitUtils.

E. util.BitUtils.process(bytes);

Answer: E

67. Given a class Repetition:

1. package utils;

2.

3. public class Repetition {

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

5. }

and given another class Demo:

1. // insert code here

2.

3. public class Demo {

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

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

6. }

7. }

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

"pizzapizza"?

A. import static utils.Repetition.twice;

B. static import utils.\*;

C. static import utils.Repetition.\*;

D. import utils.Repetition.\*;

E. import utils.\*;

F. import utils.Repetition.twice();

G. static import utils.Repetition.twice;

Answer: A

68. 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 13 to:

public Sub() { super(a); }

B. Change line 13 to:

public Sub() { this(5); }

C. Change line 13 to:

public Sub() { super(5); }

D. Change line 2 to:

protected int a;

E. Change line 2 to:

public int a;

Answer: B,C

69. Given:

10. interface Foo {}

11. class Alpha implements Foo {}

12. class Beta extends Alpha {}

13. class Delta extends Beta {

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

15. Beta x = new Beta();

16. // insert code here

17. }

18. }

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

A. Beta b = (Beta)(Alpha)x;

B. Foo f = (Alpha)x;

C. Alpha a = x;

D. Foo f = (Delta)x;

Answer: D

70. Given:

11. public abstract class Shape {

12. private int x;

13. private int y;

14. public abstract void draw();

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

16. this.x = x;

17. this.y = y;

18. }

19. }

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

A. public abstract class Circle implements Shape {

private int radius;

public void draw();

}

B. public abstract class Circle implements Shape {

private int radius;

public void draw() { /\* code here \*/ }

C. public class Circle extends Shape {

private int radius;

public void draw() {/\* code here \*/}

D. public class Circle implements Shape {

private int radius;

}

E. public abstract class Circle extends Shape {

private int radius;

}

F. public class Circle extends Shape {

private int radius;

public void draw();

}

Answer: C,E

71. Given:

1. interface TestA { String toString(); }

2. public class Test {

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

4. System.out.println(new TestA() {

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

6. });

7. }

8. }

What is the result?

A. Compilation fails because of an error in line 1.

B. null

C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 5.

E. test

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

Answer: E

01. Given:

10. class One {

11. void foo() { }

12. }

13. class Two extends One {

14. //insert method here

15. }

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

three.)

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

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

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

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

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

Answer: A,C,E

72. Given:

11. public void genNumbers() {

12. ArrayList numbers = new ArrayList();

13. for (int i=0; i<10; i++) {

14. int value = i \* ((int) Math.random());

15. Integer intObj = new Integer(value);

16. numbers.add(intObj);

17. }

18. System.out.println(numbers);

19. }

Which line of code marks the earliest point that an object referenced by intObj becomes a

candidate for garbage collection?

A. Line 19

B. The object is NOT a candidate for garbage collection.

C. Line 17

D. Line 16

E. Line 18

Answer: A

73. 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 Jumper

B. Cat is-a Animal

C. Dog is-a Jumper

D. Dog is-a Animal

E. Beagle has-a Jumper

F. Cat has-a Animal

G. Beagle has-a Tail

Answer: A,D,G

74. Given:

1. class SuperClass {

2. public A getA() {

3. return new A();

4. }

5. }

6. class SubClass extends SuperClass {

7. public B getA(){

8. return new B();

9. }

10. }

Which statement is true?

A. Compilation will always fail because of an error in line 8.

B. Compilation will succeed if A extends B.

C. Compilation will succeed if B extends A.

D. Compilation will always fail because of an error in line 7.

Answer: C

75. Given:

1. public class Plant {

2. private String name;

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

4. public String getName() { return name; }

5. }

1. public class Tree extends Plant {

2. public void growFruit() { }

3. public void dropLeaves() { }

4. }

Which statement is true?

A. The code will compile if public Plant() { this("fern"); } is added to the Plant class.

B. The code will compile if public Plant() { Tree(); } is added to the Plant class.

C. The code will compile if public Plant() { Plant("fern"); } is added to the Plant class.

D. The code will compile without changes.

E. The code will compile if public Tree() { Plant(); } is added to the Tree class.

Answer: A

76. Given:

10. class One {

11. public One() { System.out.print(1); }

12. }

13. class Two extends One {

14. public Two() { System.out.print(2); }

15. }

16. class Three extends Two {

17. public Three() { System.out.print(3); }

18. }

19. public class Numbers{

20. public static void main( String[] argv ) { new Three(); }

21. }

What is the result when this code is executed?

A. 1

B. 321

C. 3

D. The code runs with no output.

E. 123

Answer: E

77. 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:

protected int a;

B. Change line 13 to:

public Sub() { super(a); }

C. Change line 13 to:

public Sub() { this(5); }

D. Change line 13 to:

public Sub() { super(5); }

E. Change line 2 to:

public int a;

Answer: C,D

78. Given:

11. class Cup { }

12. class PoisonCup extends Cup { }

...

21. public void takeCup(Cup c) {

22. if (c instanceof PoisonCup) {

23. System.out.println("Inconceivable!");

24. } else if (c instanceof Cup) {

25. System.out.println("Dizzying intellect!");

26. } else {

27. System.exit(0);

28. }

29. }

And the execution of the statements:

Cup cup = new PoisonCup();

takeCup(cup);

What is the output?

A. An exception is thrown at runtime.

B. Compilation fails because of an error in line 22.

C. Dizzying intellect!

D. Inconceivable!

E. The code runs with no output.

Answer: D

79. Given:

11. class A {

12. public void process() { System.out.print("A,"); }

13. class B extends A {

14. public void process() throws IOException {

15. super.process();

16. System.out.print("B,");

17. throw new IOException();

18. }

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

20. try { new B().process(); }

21. catch (IOException e) { System.out.println("Exception"); }}

What is the result?

A. Compilation fails because of an error in line 14.

B. Exception

C. A,B,Exception

D. Compilation fails because of an error in line 20.

E. A NullPointerException is thrown at runtime.

Answer: A

80. Given:

31. // some code here

32. try {

33. // some code here

34. } catch (SomeException se) {

35. // some code here

36. } finally {

37. // some code here

38. }

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

A. The code on line 31 throws an exception.

B. The code on line 35 throws an exception.

C. The code on line 33 executes successfully.

D. The code on line 33 throws an exception.

E. The instance gets garbage collected.

Answer: B,C,D

81. Given:

10. interface Foo { int bar(); }

11. public class Sprite {

12. public int fubar( Foo foo ) { return foo.bar(); }

13. public void testFoo() {

14. fubar(

15. // insert code here

16. );

17. }

18. }

Which code, inserted at line 15, allows the class Sprite to compile?

A. new Foo { public int bar() { return 1; }

B. new Foo() { public int bar() { return 1; }

C. Foo { public int bar() { return 1; }

D. new class Foo { public int bar() { return 1; }

Answer: B

82. Given:

1. class TestA {

2. public void start() { System.out.println("TestA"); }

3. }

4. public class TestB extends TestA {

5. public void start() { System.out.println("TestB"); }

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

7. ((TestA)new TestB()).start();

8. }

9. }

What is the result?

A. TestB

B. An exception is thrown at runtime.

C. TestA

D. Compilation fails.

Answer: A

83. Given:

11. class Animal { public String noise() { return "peep"; } }

12. class Dog extends Animal {

13. public String noise() { return "bark"; }

14. }

15. class Cat extends Animal {

16. public String noise() { return "meow"; }

17. }

...

30. Animal animal = new Dog();

31. Cat cat = (Cat)animal;

32. System.out.println(cat.noise());

What is the result?

A. An exception is thrown at runtime.

B. peep

C. bark

D. meow

E. Compilation fails.

Answer: A

84. Given:

1. class TestA {

2. public void start() { System.out.println("TestA"); }

3. }

4. public class TestB extends TestA {

5. public void start() { System.out.println("TestB"); }

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

7. ((TestA)new TestB()).start();

8. }

9. }

What is the result?

A. TestB

B. Compilation fails.

C. TestA

D. An exception is thrown at runtime.

Answer: A

85. Given:

1. public class Plant {

2. private String name;

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

4. public String getName() { return name; }

5. }

1. public class Tree extends Plant {

2. public void growFruit() { }

3. public void dropLeaves() { }

4. }

Which statement is true?

A. The code will compile if public Tree() { Plant(); } is added to the Tree class.

B. The code will compile if public Plant() { this("fern"); } is added to the Plant class.

C. The code will compile without changes.

D. The code will compile if public Plant() { Tree(); } is added to the Plant class.

E. The code will compile if public Plant() { Plant("fern"); } is added to the Plant class.

Answer: B

86. Given:

11. class ClassA {}

12. class ClassB extends ClassA {}

13. class ClassC extends ClassA {}

and:

21. ClassA p0 = new ClassA();

22. ClassB p1 = new ClassB();

23. ClassC p2 = new ClassC();

24. ClassA p3 = new ClassB();

25. ClassA p4 = new ClassC();

Which three are valid? (Choose three.)

A. p0 = p1;

B. p1 = (ClassB)p3;

C. p2 = (ClassC)p1;

D. p2 = (ClassC)p4;

E. p2 = p4;

F. p1 = p2;

Answer: A,B,D

87. Given:

10. public class SuperCalc {

11. protected static int multiply(int a, int b) { return a \* b;}

12. }

and:

20. public class SubCalc extends SuperCalc{

21. public static int multiply(int a, int b) {

22. int c = super.multiply(a, b);

23. return c;

24. }

25. }

and:

30. SubCalc sc = new SubCalc ();

31. System.out.println(sc.multiply(3,4));

32. System.out.println(SubCalc.multiply(2,2));

What is the result?

A. The code runs with no output.

B. 12

4

C. Compilation fails because of an error in line 31.

D. Compilation fails because of an error in line 21.

E. An exception is thrown at runtime.

F. Compilation fails because of an error in line 22.

Answer: F

88. Which two statements are true about has-a and is-a relationships? (Choose two.)

A. Interfaces must be used when creating a has-a relationship.

B. Instance variables can be used when creating a has-a relationship.

C. Inheritance represents a has-a relationship.

D. Inheritance represents an is-a relationship.

Answer: B,D

89. Given:

10. interface Foo {}

11. class Alpha implements Foo {}

12. class Beta extends Alpha {}

13. class Delta extends Beta {

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

15. Beta x = new Beta();

16. // insert code here

17. }

18. }

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

A. Beta b = (Beta)(Alpha)x;

B. Alpha a = x;

C. Foo f = (Alpha)x;

D. Foo f = (Delta)x;

Answer: D

90. Given:

1. public interface A {

2. String DEFAULT\_GREETING = "Hello World";

3. public void method1();

4. }

A programmer wants to create an interface called B that has A as its parent. Which interface

declaration is correct?

A. public interface B extends A {}

B. public interface B inheritsFrom A {}

C. public interface B implements A {}

D. public interface B instanceOf A {}

Answer: A

91. Given:

11. class A {

12. public void process() { System.out.print("A,"); }

13. class B extends A {

14. public void process() throws IOException {

15. super.process();

16. System.out.print("B,");

17. throw new IOException();

18. }

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

20. try { new B().process(); }

21. catch (IOException e) { System.out.println("Exception"); }

What is the result?

A. A NullPointerException is thrown at runtime.

B. A,B,Exception

C. Compilation fails because of an error in line 14.

D. Compilation fails because of an error in line 20.

E. Exception

Answer: C

92. Which two statements are true about has-a and is-a relationships? (Choose two.)

A. Interfaces must be used when creating a has-a relationship.

B. Instance variables can be used when creating a has-a relationship.

C. Inheritance represents an is-a relationship.

D. Inheritance represents a has-a relationship.

Answer: B,C

93. Given:

11. interface DeclareStuff {

12. public static final int EASY = 3;

13. void doStuff(int t); }

14. public class TestDeclare implements DeclareStuff {

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

16. int x = 5;

17. new TestDeclare().doStuff(++x);

18. }

19. void doStuff(int s) {

20. s += EASY + ++s;

21. System.out.println("s " + s);

22. }

23. }

What is the result?

A. s 10

B. s 16

C. An exception is thrown at runtime.

D. Compilation fails.

E. s 14

Answer: D

94. Given:

10. interface Data { public void load(); }

11. abstract class Info { public abstract void load(); }

Which class correctly uses the Data interface and Info class?

A. public class Employee extends Info implements Data{

public void Data.load() { /\*do something\*/ }

public void Info.load() { /\*do something\*/ }

}

B. public class Employee extends Info implements Data

public void load(){ /\*do something\*/ }

public void Info.load(){ /\*do something\*/ }

}

C. public class Employee extends Info implements Data {

public void load() { /\*do something\*/ }

}

D. public class Employee implements Info extends Data {

public void Data.load(){ /\*do something\*/ }

public void load(){ /\*do something\*/ }

}

E. public class Employee implements Info extends Data {

public void load() { /\*do something\*/ }

}

F. public class Employee implements Info extends Data {

public void load(){ /\*do something\*/ }

public void Info.load(){ /\*do something\*/ }

}

Answer: C

95. Given:

1. class ClassA {

2. public int numberOfInstances;

3. protected ClassA(int numberOfInstances) {

4. this.numberOfInstances = numberOfInstances;

5. }

6. }

7. public class ExtendedA extends ClassA {

8. private ExtendedA(int numberOfInstances) {

9. super(numberOfInstances);

10. }

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

12. ExtendedA ext = new ExtendedA(420);

13. System.out.print(ext.numberOfInstances);

14. }

15. }

Which statement is true?

A. An exception is thrown at runtime.

B. All constructors must be declared public.

C. Constructors CANNOT use the private modifier.

D. 420 is the output.

E. Constructors CANNOT use the protected modifier.

Answer: D

96. Given:

10. abstract public class Employee {

11. protected abstract double getSalesAmount();

12. public double getCommision() {

13. return getSalesAmount() \* 0.15;

14. }

15. }

16. class Sales extends Employee {

17. // insert method here

18. }

Which two methods, inserted independently at line 17, correctly complete the Sales class?

(Choose two.)

A. private double getSalesAmount() { return 1230.45; }

B. protected double getSalesAmount() { return 1230.45; }

C. public double getSalesAmount() { return 1230.45; }

D. double getSalesAmount() { return 1230.45; }

Answer: B,C

97. 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. Compilation fails.

B. 42042

C. An exception is thrown at runtime.

D. 462

E. 42

F. 420

Answer: B

98. Given:

33. try {

34. // some code here

35. } catch (NullPointerException e1) {

36. System.out.print("a");

37. } catch (RuntimeException e2) {

38. System.out.print("b");

39. } finally {

40. System.out.print("c");

41. }

What is the result if a NullPointerException occurs on line 34?

A. ab

B. abc

C. bc

D. c

E. ac

F. a

Answer: E

99. Given:

10: public class Hello {

11: String title;

12: int value;

13: public Hello() {

14: title += " World";

15: }

16: public Hello(int value) {

17: this.value = value;

18: title = "Hello";

19: Hello();

20: }

21: }

and:

30: Hello c = new Hello(5);

31: System.out.println(c.title);

What is the result?

A. An exception is thrown at runtime.

B. Hello World 5

C. The code runs with no output.

D. Hello World

E. Compilation fails.

F. Hello

Answer: E

100. Click the Exhibit button.

Given this code from Class B:

25. A a1 = new A();

26. A a2 = new A();

27. A a3 = new A();

28. System.out.println(A.getInstanceCount());

What is the result?

A. Compilation of class A fails.

B. Line 28 prints the value 3 to System.out.

C. Line 28 prints the value 1 to System.out.

D. Compilation fails because of an error on line 28.

E. A runtime error occurs when line 25 executes.

Answer: A

101. Given

11. public interface Status {

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

13. }

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

A. static

B. native

C. abstract

D. final

E. private

F. protected

G. public

Answer: A,D,G

102. Given:

10. interface A { void x(); }

11. class B implements A { public void x() {} public void y() {} }

12. class C extends B { public void x() {} }

And:

20. java.util.List<A> list = new java.util.ArrayList<A>();

21. list.add(new B());

22. list.add(new C());

23. for (A a : list) {

24. a.x();

25. a.y();

26. }

What is the result?

A. An exception is thrown at runtime.

B. The code runs with no output.

C. Compilation fails because of an error in line 25.

D. Compilation fails because of an error in line 20.

E. Compilation fails because of an error in line 21.

F. Compilation fails because of an error in line 23.

Answer: C

103. Given:

11. static class A {

12. void process() throws Exception { throw new Exception(); }

13. }

14. static class B extends A {

15. void process() { System.out.println("B "); }

16. }

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

18. A a = new B();

19. a.process();

18. Given:

12. System.out.format("Pi is approximately %d.", Math.PI);

What is the result?

A. An exception is thrown at runtime.

B. Pi is approximately 3.

C. Pi is approximately 3.141593.

D. Compilation fails.

Answer: A

104. 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

Answer: C

105. Given:

11. static class A {

12. void process() throws Exception { throw new Exception(); }

13. }

14. static class B extends A {

15. void process() { System.out.println("B "); }

16. }

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

18. A a = new B();

19. a.process();

20. }

What is the result?

A. Compilation fails because of an error in line 19.

B. An exception is thrown at runtime.

C. B

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

E. Compilation fails because of an error in line 15.

F. The code runs with no output.

Answer: A

106. What is the result?

A. The code runs with no output.

B. Compilation fails because of an error in line 15.

C. Compilation fails because of an error in line 19.

D. An exception is thrown at runtime.

E. B

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

Answer: C

107. Click the Exhibit button.

Given:

25. try {

26. A a = new A();

27. a.method1();

28. } catch (Exception e) {

29. System.out.print("an error occurred");

30. }

Which two statements are true if a NullPointerException is thrown on line 3 of class C? (Choose

two.)

A. The application will crash.

B. The code on line 29 will be executed.

C. The code on line 5 of class A will execute.

D. The exception will be propagated back to line 27.

E. The code on line 5 of class B will execute.

Answer: B,D

108 Given:

11. Float pi = new Float(3.14f);

12. if (pi > 3) {

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

14. }

15. else {

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

17. }

18. finally {

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

20. }

What is the result?

A. pi is not bigger than 3. Have a nice day.

B. pi is bigger than 3. Have a nice day.

C. Compilation fails.

D. pi is bigger than 3.

E. An exception occurs at runtime.

Answer: C

109. try { test(); }

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

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

19. }

What is the result?

A. exception end

B. exception test end

C. end

D. A Throwable is thrown by main.

E. Compilation fails.

F. An Exception is thrown by main.

Answer: D

110. Given:

11. static void test() {

12. try {

13. String x = null;

14. System.out.print(x.toString() + " ");

15. }

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

17. }

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

19. try { test(); }

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

21. }

What is the result?

A. finally exception

B. Compilation fails.

C. null finally

D. finally

E. null

Answer: A

111. Given:

11. static void test() throws RuntimeException {

12. try {

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

14. throw new RuntimeException();

15. }

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

17. }

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

19. try { test(); }

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

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

22. }

What is the result?

A. test runtime end

B. A Throwable is thrown by main at runtime.

C. test end

D. Compilation fails.

E. test exception end

Answer: E

112. Given:

12. import java.io.\*;

13. public class Forest implements Serializable {

14. private Tree tree = new Tree();

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

16. Forest f = new Forest();

17. try {

18. FileOutputStream fs = new FileOutputStream("Forest.ser");

19. ObjectOutputStream os = new ObjectOutputStream(fs);

20. os.writeObject(f); os.close();

21. } catch (Exception ex) { ex.printStackTrace(); }

22. } }

23.

24. class Tree { }

What is the result?

A. An instance of Forest is serialized.

B. Compilation fails.

C. An exception is thrown at runtime.

D. An instance of Forest and an instance of Tree are both serialized.

Answer: C

113. Given:

11. public static void parse(String str) {

12. try {

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

14. } catch (NumberFormatException nfe) {

15. f = 0;

16. } finally {

17. System.out.println(f);

18. }

19. }

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

21. parse("invalid");

22. }

What is the result?

A. A NumberFormatException is thrown by the parse method at runtime.

B. A ParseException is thrown by the parse method at runtime.

C. Compilation fails.

D. 0.0

Answer: C

114. Given:

33. try {

34. // some code here

35. } catch (NullPointerException e1) {

36. System.out.print("a");

37. } catch (RuntimeException e2) {

38. System.out.print("b");

39. } finally {

40. System.out.print("c");

41. }

What is the result if a NullPointerException occurs on line 34?

A. c

B. a

C. abc

D. bc

E. ac

F. ab

Answer: E

115. Given:

31. // some code here

32. try {

33. // some code here

34. } catch (SomeException se) {

35. // some code here

36. } finally {

37. // some code here

38. }

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

A. The code on line 33 executes successfully.

B. The code on line 33 throws an exception.

C. The instance gets garbage collected.

D. The code on line 31 throws an exception.

E. The code on line 35 throws an exception.

Answer: A,B,E

116. Given:

12. import java.io.\*;

13. public class Forest implements Serializable {

14. private Tree tree = new Tree();

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

16. Forest f = new Forest();

17. try {

18. FileOutputStream fs = new FileOutputStream("Forest.ser");

19. ObjectOutputStream os = new ObjectOutputStream(fs);

20. os.writeObject(f); os.close();

21. } catch (Exception ex) { ex.printStackTrace(); }

22. } }

23.

24. class Tree { }

What is the result?

A. Compilation fails.

B. An instance of Forest is serialized.

C. An exception is thrown at runtime.

D. An instance of Forest and an instance of Tree are both serialized.

Answer: C

117. Given:

12. import java.io.\*;

13. public class Forest implements Serializable {

14. private Tree tree = new Tree();

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

16. Forest f = new Forest();

17. try {

18. FileOutputStream fs = new FileOutputStream("Forest.ser");

19. ObjectOutputStream os = new ObjectOutputStream(fs);

20. os.writeObject(f); os.close();

21. } catch (Exception ex) { ex.printStackTrace(); }

22. } }

23.

24. class Tree { }

What is the result?

A. An instance of Forest and an instance of Tree are both serialized.

B. An instance of Forest is serialized.

C. Compilation fails.

D. An exception is thrown at runtime.

Answer: D

118. Given:

12. import java.io.\*;

13. public class Forest implements Serializable {

14. private Tree tree = new Tree();

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

16. Forest f = new Forest();

17. try {

18. FileOutputStream fs = new FileOutputStream("Forest.ser");

19. ObjectOutputStream os = new ObjectOutputStream(fs);

20. os.writeObject(f); os.close();

21. } catch (Exception ex) { ex.printStackTrace(); }

22. } }

23.

24. class Tree { }

What is the result?

A. An instance of Forest and an instance of Tree are both serialized.

B. An instance of Forest is serialized.

C. An exception is thrown at runtime.

D. Compilation fails.

Answer: C

119. Which four statements are true? (Choose four.)

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

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

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

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

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

F. Has-a relationships should never be encapsulated.

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

Answer: A,C,D,G

1. What type of inheritance does Java have?

a. single inheritance

b. double inheritance

c. multiple inheritance

d. class inheritance

Ans: A

2. Say that there are three classes: Computer, AppleComputer, and IBMComputer. What are the likely relationships between these classes?

a. Computer is the superclass, AppleComputer and IBMComputer are subclasses of Computer.

b. IBMComputer is the superclass, AppleComputer and Computer are subclasses of IBMComputer.

c. Computer, AppleComputer and IBMComputer are sibling classes.

d. Computer is a superclass, AppleComputer is a subclasses of Computer, and IBMComputer is a subclass of AppleComputer

Ans: A

3. Can an object be a subclass of another object?

a. Yes—as long as single inheritance is followed.

b. No—inheritance is only between classes.

c. Only when one has been defined in terms of the other.

d. Yes—when one object is used in the constructor of another.

Ans: B

4. How many objects of a given class can there be in a program?

a. One per defined class.

b. One per constructor definition

c. As many as the program needs.

d. One per main() method

Ans: C

5. What restriction is there on using the super reference in a constructor?

a. It can only be used in the parent's constructor.

b. Only one child class can use it.

c. It must be used in the last statement of the constructor.

d. It must be used in the first statement of the constructor.

Ans: D

6. Which of the following is correct syntax for defining a new class Jolt based on the superclass SoftDrink?

a. class Jolt isa SoftDrink { //additional definitions go here }

b. class Jolt implements SoftDrink { //additional definitions go here }

c. class Jolt defines SoftDrink { //additional definitions go here }

d. class Jolt extends SoftDrink { //additional definitions go here }

Ans: D

7. A class Car and its subclass Yugo both have a method run() which was written by the programmer as part of the class definition. If junker refers to an object of type Yugo, what will the following code do? junker.show();

a. The show() method defined in Yugo will be called

b. The show() method defined in Car will be called.

c. The compiler will complain that run() has been defined twice.

d. Overloading will be used to pick which run() is called.

Ans: A

8. A class Animal has a subclass Mammal. Which of the following is true:

a. Because of single inheritance, Mammal can have no subclasses.

b. Because of single inheritance, Mammal can have no other parent than Animal.

c. Because of single inheritance, Animal can have only one subclass.

d. Because of single inheritance, Mammal can have no siblings.

Ans: B

9. Does a subclass inherit both member variables and methods?

a. No—only member variables are inherited.

b. No—only methods are inherited.

c. Yes—both are inherited.

d. Yes—but only one or the other are inherited.

Ans: C

10. Which of the following is NOT an advantage to using inheritance?

a. Code that is shared between classes needs to be written only once.

b. Similar classes can be made to behave consistently.

c. Enhancements to a base class will automatically be applied to derived classes.

d. One big superclass can be used instead of many little classes.

Ans D

11. Given:

public class Base {

public void aMethod() throws ClassNotFoundException { }

}

public class Derived extends Base {

public void aMethod() throws RuntimeException { }

}

Assuming that the classes are in two seperate files, compilation of the Dervied.java causes

a. A compiler error because RuntimeException is not a subclass if ClassNotFoundException.

b. No compiler error.

Ans: B

12. Interface methods can be declared with the following modifiers

a. public

b. none (i.e., no access modifier).

c. private.

d. static

e. native

f. synchronized.

Ans: A, B

13. What is an abstract class?

a. An abstract class is one without any child classes.

b. An abstract class is any parent class with more than one child class.

c. An abstract class is class which cannot be instantiated.

d. abstract class is another name for "base class."

Ans: C

14. What is an abstract method?

a. An abstract method is any method in an abstract class

b. An abstract method is a method which cannot be inherited.

c. An abstract method is one without a body that is declared with the reserved word abstract

d. An abstract method is a method in the child class that overrids a parent method.

Ans: C

15. Can an abstract class define both abstract methods and non-abstract methods?

a. No—it must have all one or the other

b. No—it must have all abstract methods.

c. Yes—but the child classes do not inherit the abstract methods.

d. Yes—the child classes inherit both.

Ans: D

16. Can an abstract parent class have non-abstract children?

a. No—an abstract parent must have only abstract children.

b. No—an abstract parent must have no children at all.

c. Yes—all children of an abstract parent must be non-abstract.

d. Yes—an abstract parent can have both abstract and non-abstract children

Ans: D

17. Can an abstract method be defined in a non-abstract class?

a. No—if a class defines an abstract method the class itself must be abstract

b. No—only classes are abstract, not methods

c. Yes—a method can be declared abstract in any parent as long as the child classes also declare it abstract.

d. Yes—there is no restriction on where abstract methods can be defined.

Ans: A

18. Can an object of a child type be assigned to a variable of the parent type? For

example,

Card crd;

BirthDay bd = new BirthDay("Lucinda", 42);

crd = bd; // is this correct?

a. No—there must always be an exact match between the variable and the object types.

b. No—but a object of parent type can be assigned to a variable of child type.

c. Yes—an object can be assigned to a reference variable of the parent type.

d. Yes—any object can be assigned to any reference variable.

Ans: C

19. What is polymorphism in Java?

a. It is when a single variable is used with several different types of related objects at different places in a program.

b. It is when a program uses several different types of objects, each with its own variable.

c. It is when a single parent class has many child classes.

d. It is when a class has several methods with the same name but different parameter types.

Ans A

20. In order for the following code to be correct, what must be the type of the reference variable card?

\_\_\_\_\_\_\_\_\_ card;

card = new Valentine( "Joe", 14 ) ;

card.greeting();

card = new Holiday( "Bob" ) ;

card.greeting();

card = new Birthday( "Emily", 12 ) ;

card.greeting();

a. Valentine

b. Holiday

c. Birthday

d. Card

Ans: D

21. What is an advantage of polymorphism?

a. The same program logic can be used with objects of several related types.

b. Variables can be re-used in order to save memory.

c. Constructing new objects from old objects of a similar type saves time.

d. Polymorphism is a dangerous aspect of inheritance and should be avoided.

Ans: A

22. Read this piece of code carefully

if("String".toString() == "String")

System.out.println("Equal");

else

System.out.println("Not Equal");

a. the code will compile an print “Equal”.

b. the code will compile an print “Not Equal”.

c. the code will cause a compiler error.

Ans: A

23. Read this piece of code carefully

System.out.println("String".substring(0,4));

a. the code will print "Strin" on the screen.

b. the code will print "Stri" on the screen.

c. the code will cause a compiler error.

Ans: B

24. Read this piece of code carefully

if("String".replace('g','G') == "String".replace('g','G'))

System.out.println("Equal");

else

System.out.println("Not Equal");

a. the code will compile an print "Equal".

b. the code will compile an print "Not Equal".

c. the code will cause a compiler error

Ans B

25. Read this piece of code carefully

if( "STRING".toUpperCase() == "STRING")

System.out.println("Equal");

else

System.out.println("Not Equal");

a. the code will compile an print "Equal".

b. the code will compile an print "Not Equal".

c. the code will cause a compiler error

Ans: A

26. Read this piece of code carefully

if("String".substring(0) == "String")

System.out.println("Equal");

else

System.out.println("Not Equal");

a. the code will compile an print "Equal".

b. the code will compile an print "Not Equal".

c. the code will cause a compiler error

Ans: A

27. A String Class

a. is final

b. is public

c. is serializable

d. has a constructor which takes a StingBuffer Object as an Argument

Ans A, B, C, D

28. Read this piece of code carefully

if("String".trim() == "String")

System.out.println("Equal");

else

System.out.println("Not Equal");

a. the code will compile an print “Equal”.

b. the code will compile an print “Not Equal”.

c. the code will cause a compiler error

Ans: B

29. Read the code below. Will be the result of attempting to compile and run the code below.

public class AQuestion {

public void method(Object o){

System.out.println("Object Verion");

}

public void method(String s){

System.out.println("String Version");

}

public static void main(String args[]){

AQuestion question = new AQuestion();

question.method(null);

}

}

a. The code does not compile.

b. The code compiles cleanly and shows “Object Version”.

c. The code compiles cleanly and shows “String Version”

d. The code throws an Exception at Runtime.

Ans C

30. Read the code below. Will be the result of attempting to compile and run the code below.

public class AQuestion{

public void method(StringBuffer sb){

System.out.println("StringBuffer Verion");

}

public void method(String s){

System.out.println("String Version");

}

public static void main(String args[]){

AQuestion question = new AQuestion();

question.method(null);

}

}

a. The code does not compile.

b. The code compiles cleanly and shows “StringBuffer Version”.

c. The code compiles cleanly and shows “String Version”

d. The code throws an Exception at Runtime

Ans: A

31. Read the code below carefully.

public class TestClass {

public static void main(String Args[]){

StringBuffer sb1 = new StringBuffer("String");

StringBuffer sb2 = new StringBuffer("String");

if(sb1.equals(sb2)){

//lots of code

}

}

}

Is the line marked "lots of code" ever reached?

a. Yes

b. No

Ans: B

32. Read the following code carefully.

public class AStringQuestion {

static String s1;

static String s2;

public static void main(String args[]) {

s2 = s1+s2;

System.out.println(s2);

}

}

Attempting to compile and run the code

a. Will cause a compilation error.

b. Runtime Execption - NullPointerException in the 2nd line of the main method.

c. Will compile successfully and print nullnull on the screen.

d. Will compile successfully and print an empty line on the screen.

e. Will compile successfully and print nothing on the screen.

Ans: C

33. public class A {

final StringBuffer sb = new StringBuffer("I am final");

public A(){

}

public StringBuffer getSb() {

return this.sb;

}

public static void main(String[] args){

A a = new A();

StringBuffer localSB = a.getSb();

localSB.append("....");

localSB = new StringBuffer();

System.out.println(localSB.toString());

}

}

Attempting to compile and run the above application will yield

a. A compilation error : Final variable being assigned to a non-final handle.

b. A compilation error : implicitly final localSB being reassigned

c. A Runtime Error : Attempted reassignment to a final handle (localSB)

d. No Errors during compilation and execution. An Empty line is printed on the console.

Ans: D

34. Which of the following are true about the class defined inside an interface

a. it is not possible in the java Laungage.

b. The class is always public.

c. The class is always static.

d. the class methods cannot call the methods declared in the interface.

e. the class methods can call only the static methods declared in the interface.

Ans: B, C, D

35. Read the following code below.

public interface AQuestion{

public abstract void someMethod() throws Exception;

}

A Class implementing this interface should

a. Necessarily be an abstract class.

b. Should have the method public abstract void someMethod();

c. Should have the method public void someMethod() which has to throw an exception which is a subclass of java.lang.Exception.

d. Should have the method public void someMethod() which need not throw an Exception.

Ans D

36. An Interface can never be private or protected.

a. True

b. False

Ans: B

37. public interface AQuestion{

void someMethod();

}

The class which implements AQuestion

a. Should have someMethod which must necessarily be public.

b. Should have someMethod which could be “friendly” or public

c. Should have someMethod which should not throw any checked exceptions.

d. Should have someMethod which cannot be sychronized as sychronized is not in the signature of the interface defination

Ans A, C

38. public class AQuestion{

private int i = j;

private int j = 10;

public static void main(String args[]){

System.out.println((new AQuestion()).i);

}

}

a. Compiler error complaining about access restriction of private variables of AQuestion.

b. Compiler error complaining about forward referencing.

c. No error - The output is 0;

d. No error - The output is 10;

Ans: B

39. public class AQuestion {

private int i = giveMeJ();

private int j = 10;

private int giveMeJ(){

return j;

}

public static void main(String args[]){

System.out.println((new AQuestion()).i);

}

}

a. Compiler error complaining about access restriction of private variables of AQuestion.

b. Compiler error complaining about forward referencing.

c. No Compilation error - The output is 0;

d. No Compilation error - The output is 10;

Ans: C

40. Given

public XXXX extends something1, something2

a. XXX should be an interface,something1 and something2 need not, for the expression to be legal

b. XXX should be a class, something1 and something2 must be interfaces for the expression to be legal.

c. XXX, something1 and something2 must be interfaces for the expression to be legal.

d. The above statement is alway illegal in Java as multiple inheritance is not supported.

Ans C

41. interface One {

public void someMethod();

}

public class One\_impl implements One {

public native void someMethod();

}

Assuming that the native method is not provided in any local library, an attempt to compile and run the above lines of code will cause

a. Compilation error - implimentations can never be native.

b. Compilation error - method not found in local libraries.

c. Runtime Exception - method not found in local libraries.

d. Compilation successfull but runtime error is thrown if and only if the method someMethod of class One\_impl is called.

Ans: D

42. public class AQuestion {

public static void main(String args[]){

System.out.println("Before Try");

try { }

catch(Throwable t) {

System.out.println("Inside Catch");

}

System.out.println("At the End");

}

}

a. Compiler error complaining about the catch block, where no Throwable object can ever be thrown.

b. Compiler error - Throwable Object can not be caught, only Exceptions must be caught.

c. No compiler error. The lines "Before Try" and "At the end" are printed on the screen.

Ans: C

43. public class AQuestion {

public static void main(String args[]) {

System.out.println("Before Try");

try { }

catch(java.io.IOException t) {

System.out.println("Inside Catch");

}

System.out.println("At the End");

}

}

a. Compiler error complaining about the catch block where no IOException object can ever be thrown.

b. Compiler error - IOException not found. It must be imported in the first line of the code.

c. No compiler error. The lines "Before Try" and "At the end" are printed on the screen.

Ans A

44. The class java.lang.Exception

a. Is public

b. Extends Throwable

c.Implements Throwable

d.Is serializable

Ans: A, B, D

45. Which of the following statements are true?

a. A method can throw an Exception

b. A method can return an Exception

Ans A, B

46. public class A {

private void method1() throws Exception {

throw new RuntimeException();

}

public void method2(){

try {

method1();

}

catch(RuntimeException e) {

System.out.println("Caught Runtime Exception");

}

catch(Exception e){

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

}

}

public static void main(String args[]){

A a = new A();

a.method2();

}

}

The above lines of code -

a. will not compile.

b. will compile and show - "Caught Runtime Exception".

c. will compile and show - "Caught Exception".

d. will compile and show both the messages one after another in the order they appear.

Ans: B

47. The following lines of code

byte b = 0;

b += 1;

a. results in b having the value 1.

b. causes a compiler error.

c. will require a cast (byte) before 1.

Ans: A

48. The following express

char c = -1;

a. will cause a compiler error as the range of character is between 0 and 2^16 - 1. Will request for an explicit cast.

b. will not cause a compiler error and c will have the value -1;

c will not represent any ascii character.

d. will still be a unicode character.

Ans: A

49. All the wrapper classes (Integer, Boolean, Float, Short, Long, Double and Character)

a. are public

b. are serializable

c. are immutatable

d. extend java.lang.Number are final

Ans: B

50. public class ADirtyOne {

char a = '\u000A';

}

An attempt to compile the above class

a. will complete successfully.

b. will not compile as 0x000A is out of range for unicode charaters.

c. will complain about illegal character assignment

d. will compile but will cause a runtime error in accessing the variable.

Ans: C

51. public class ADirtyOne {

//char a = '\u000A';

}

An attempt to compile the above class

a. will complete successfully.

b. will compile sucessfully but with a warning message.

c. will not compile - complains on an invalid expression.

Ans: C

52. public class AnotherDirtyOne {

private final int i =10;

private byte k = i;

}

An attempt to compile and run the above code will

a. Cause a compilation error due to invalid assignment ( int to byte) and will request for an explicit cast to be done on i [ k=(byte) i ].

b. Compilation occurs with a warning message - suggesting that the accuracy of k is questionable

c. Compilation will occur cleanly without any warning message.

d. Runtime error occurs when accessing k.

Ans: C

53. what does the following expression return?

(0.0 == -0.0)

a. true

b. false

Ans: A

54. The no-argument constructor provided by the compiler when no constructor is explicitly provided in the code

a. is always public

b. is always "friendly"

c. always defaults to the access modifier provided for the class.

d. depends on the compilation options of javac

Ans: C

55. An Outer class which has all its constructors declared as private

a. Cannot be instantiated by any other Outer class.

b. Cannot be extended.

c. Both i and ii.

d. has to be declared final.

Ans: C

56. Read the code below carefully.

import java.io.\*;

public class OutOut {

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

PrintStream pr = new PrintStream(new FileOutputStream("outfile"));

System.out = pr;

System.out.println("Lets see what I see now??");

}

}

a. The code causes a compiler error. out is a declared final in System and cannot be assigned to pr.

b. The code causes a runtime Exception due the assignment to a final variable.

c. The code compiles and runs success fully. A file called "outfile" is created and "Lets see what I see now??" is printed in the same.

Ans: A

57. If the finalize() method of an object is re-references an object so that it becomes in-eligible for garbage collection

a. The compiler will throw an error.

b. The garbage collector "collects" the object anyway.

c. The garbage collector does not collect the object in the present sweep. But when the object becomes eligible for garbage collection

again, its finalize method will not be called by the garbage collector (i.e., if the garbage collector gets a chance to run.). It will simply be garbage collected.

d. The object can never be garbage collected and hence leads to memory-leak. Each time the garbage collector calls finalize before "collecting" the object the object "resurrects" itself.

Ans: C

58. What is the base type of the array given below?

Color[] example = new Color[20];

a. Color

b. example

c. No base type

d. Both (A) and (B)

Ans: A

60. What is the difference between

String rats;

and

String[] rats; ?

a. There is no difference; both declare rats to be a reference variable

b. The first declares rats to be a reference to a String object, the second declares rats to be a reference to an array of String references

c. The first constructs a single String object; the second constructs an array of String objects.

d. The first initializes rats to null; the second initializes rats to an array of nulls

Ans: B

61. What does the following statement do?

int[] values = new int[10] ;

a.It declares values to be a reference to an array object and constructs an array object containing 10 integers which are initialized to zero.

b.It declares values to be a reference to an array object, but initializes it to null

c.It declares values to be a reference to an array object which does not yet exist,

but will contain 10 zeros when it does

d.It declares values to be a reference to an array which contains 10 references to int variables.

Ans: A

62. Given the declaration

String[] names = new String[10] ;

Which of the following statements puts a reference to the String "Hello" in the last slot of the array?

a. names[0] = "Hello" ;

b. names[10] = "Hello" ;

c. names[9] = "Hello" ;

d. String[ names.length-1 ] = "Hello" ;

Ans: C

63. Say that names has been declared

String[] names = new String[10] ;

and that further statements (not shown) have put String references into some of the slots.

Which of the following fragments prints out every String, but skips null references?

a. for ( int j = 0; names[j] != null; j++ )

System.out.println( names[j] );

b. for ( int j = 0; j < names.length; j++ )

System.out.println( names[j] );

c. for ( int j = 0; j < names.length && names[j] != null ; j++ )

System.out.println( names[j] );

d. for ( int j = 0; j < names.length; j++ )

if ( names[j] != null )

System.out.println( names[j] );

Ans D

64. Say that names has been declared

String[] names = new String[10] ;

and that further statements (not shown) have put String references into some of the slots.

Which of the following fragments prints out the slots of the array from last to first, skipping slots that contain null?

a.

for ( int j = 0; j < names.length; j++ )

if ( names[j] != null )

System.out.println( names[j] );

b.

for ( int j = names.length; j < names.length; j++ )

if ( names[j] != null )

System.out.println( names[j] );

c.

for ( int j = names.length-1; j >= 0; j-- )

if ( names[j] != null )

System.out.println( names[j] );

d.

for ( int j = names.length; j >= 0; j++ )

if ( names[j] != null )

System.out.println( names[j] );

Ans: C

65. Each pass through a loop is called a/an

a. enumeration

b. iteration

c. culmination

d. pass through

Ans: B

66. Which looping process checks the test condition at the end of the loop?

a. for

b. while

c. do-while

d. no looping process checks the test condition at the end

Ans: C

67. A continue statement causes execution to skip to

a. the end of the program

b. the first statement after the loop

c. the statement following the continue statement

d. the next iteration of the loop

Ans: D

68. In a group of nested loops, which loop is executed the most number of times?

a. the outermost loop

b. the innermost loop

c. all loops are executed the same number of times

d. cannot be determined without knowing the size of the loops

Ans B

69. The statement i++; is equivalent to

a. i = i + i;

b. i = i + 1;

c. i = i - 1;

d. i - - ;

Ans: B

70. Which looping process is best used when the number of iterations is known?

a. for

b. while

c. do-while

d. all looping processes require that the iterations be known

Ans; A

71. What's wrong? for (int k = 2, k <= 12, k++)

a. the increment should always be ++k

b. the variable must always be the letter i when using a for loop

c. there should be a semicolon at the end of the statement

d. the commas should be semicolons

Ans: D

72. What's wrong? while( (i < 10) && (i > 24))

a. the logical operator && cannot be used in a test condition

b. the while loop is an exit-condition loop

c. the test condition is always false

d. the test condition is always true

Ans: C

73. Examine the following code

int count = 0;

while ( count <= 6 )

{

System.out.print( count + " " );

count = count + 2;

}

System.out.println( );

What does this code print on the monitor?

a. 1 2 3 4 5 6

b. 0 2 4 6 8

c. 0 2 4

d. 0 2 4 6

Ans: D

74. Examine the following code:

int count = 7;

while ( count >= 4 )

{

System.out.print( count + " " );

count = count - 1;

}

System.out.println( );

What does this code print on the monitor?

a. 1 2 3 4 5 6 7

b. 7 6 5 4

c. 6 5 4 3

d. 7 6 5 4 3

Ans: B

75. Examine the following code:

int count = 1;

while ( \_\_\_\_\_\_\_\_\_\_\_ )

{

System.out.print( count + " " );

count = count + 1;

}

System.out.println( );

What condition should be used so that the code writes out:

1 2 3 4 5 6 7 8

a. count < 8

b. count < 9

c. count+1 <= 8

d. count != 8

Ans: B

76. Given

for (int i = 0; i <= 3;) {

System.out.println("i = " + i);

}

a. i = 0 b. i = 0 infinitely c. The code does not run d. The code does not compile

i = 1

i = 2

i = 3

Ans: B

77. Which statements about the output of the following program are true?

public class Logic {

public static void main(String args[]) {

int i = 0;

int j = 0;

boolean t = true;

boolean r;

r = (t && 0<(i+=1));

r = (t && 0<(i+=2));

r = (t && 0<(j+=1));

r = (t || 0<(j+=2));

System.out.println( i + “ ” + j );

}

}

(a) The first digit printed is 1.

(b) The first digit printed is 2.

(c) The first digit printed is 3.

(d) The second digit printed is 1.

(e) The second digit printed is 2.

(f) The second digit printed is 3.

Answers: (c), (d)

78. Which statements about the output of the following program are true?

public static void main(String args[])

{

int i =0;

i = i++;

System.out.println(i);

}

(a) 0 is printed.

(b) 1 is printed.

Answers: (a)

79. Which statements about the output of the following program are true?

public class EqualTest {

public static void main(String args[]) {

String s1 = “YES”;

String s2 = “YES”;

if ( s1 == s2 ) System.out.println(“equal”);

String s3 = new String(“YES”);80. What will be the result of compiling and running the given program?

Select one correct answer.

1 class Q1

2 {

3 public static void main(String arg[])

4 {

5 int a[]={2,2};

6 int b=1;

7 a[b]=b=0;

8 System.out.println(a[0]);

9 System.out.println(a[1]);

10 }

11 }

(a) Compile time error at the line no. 5.

(b) Run time error at the line no. 5.

(c) Program compiles correctly and print 2,0 when executed.

(d) Program compiles correctly and print 0,2 when executed.

Answers: (c)

81. The following method is designed to convert an input string to a floating point number while detecting a bad format. [3]

I think factor must have been declared as float member variable or something]

public boolean strCvt(String s) {

try {

factor = Float.valueOf(s).floatValue();

return true;

} catch (NumberFormatException e) {

System.out.println(“Bad number ” + s);

factor = Float.NaN;

} finally {

System.out.println(“Finally”);

}

return false;

}

Which of the following descriptions of the results of various inputs to the method are correct?

(a) Input = “0.234” – Result: factor = 0.234, “Finally” is printed, true is returned.

(b) Input = “0.234” – Result: factor = 0.234, “Finally” is printed, false is returned.

(c) Input = null – Result: factor = NaN, “Finally” is printed, false is returned.

(d) Input = null – Result: factor = unchanged, “Finally” is printed, NullPointerException is thrown.

Ans: A, c

82. Here is the hierarchy of exceptions related to array index and string index errors: [3]

Exception

+-- RuntimeException

+-- IndexOutOfBoundsException

+-- ArrayIndexOutOfBoundsException

+-- StringIndexOutOfBoundsException

Suppose you had a method X that could throw both array index and string index exceptions. Assuming that X does not have any try-catch statements, which of the following statements are correct?

(a) The declaration for X must include “throws ArrayIndexOutOfBoundsException, StringIndexOutOfBoundsException”.

(b) If a method calling X catches IndexOutOfBoundsException, both array and string index exceptions will be caught.

(c) If the declaration for X includes “throws IndexOutOfBoundsException”, any calling method must use a try-catch block.

(d) The declaration for X does not have to mention exceptions.

Ans: b,c

83. Which will be the first line to cause an error in the following code? [8]

Select one correct answer.

1 class Char

2 {

3 public static void main(String arg[])

4 {

5 while(false)

6 {

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

8 }

9 while(false)

10 {

11 }

12 do;

13 while(false);

14 do

15 {

16 ;

17 }

18 while(false);

19 }

20 }

(a) Line no. 5

(b) Line no. 9

(c) Line no. 12

(d) Line no. 16

Ans: 12

84. What will be the result of compiling and running the given program? [8]

Select one correct answer.

1 public class exception

2 {

3 public static void main(String args[])

4 {

5 System.out.println("A");

6 try

7 {

8 }

9 catch(java.io.IOException t)

10 {

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

12 }

13 System.out.println("C");

14 }

15 }

(a) Compile time error.

(b) Program compiles correctly and prints "A" when executed.

(c) Program compiles correctly and prints "A" and "C" when executed.

(d) Run time error.

Ans: A

85.What will be the result of compiling and running the given program? [8]

Select one correct answer.

1 public class exception

2 {

3 public static void main(String args[])

4 {

5 System.out.println("A");

6 try

7 {

8 return;

9 }

10 catch(Exception e)

11 {

12 System.out.println("B");

13 }

14 System.out.println("C");

15 }

16 }

(a) Compile time error in line no. 8 as main() method is declared void.

(b) Program compiles correctly and prints "A" when executed.

(c) Program compiles correctly and prints "A" and "C" when executed.

(d) Compile time error at line no.14 due to statement not reached.

Ans:

86. What will be the result of compiling and running the given program? [8]

Select one correct answer.

1 public class exception

2 {

3 public static void main(String args[])

4 {

5 System.out.println("A");

6 try

7 {

8 return;

9 }

10 catch(Exception e)

11 {

12 System.out.println("B");

13 }

14 finally

15 {

16 System.out.println("C");

17 }

18 }

19 }

(a) Compile time error in line no. 8 as main() method is declared void.

(b) Program compiles correctly and prints "A" when executed.

(c) Program compiles correctly and prints "A" and "C" when executed.

(d) Program compiles correctly and prints "A","B" and "C" when executed.

Ans:

87. What will be the result of compiling and running the given program? [8]

Select one correct answer.

1 public class exception

2 {

3 public static void main(String args[])

4 {

5 System.out.println("A");

6 try

7 {

8 System.out.println("B");

9 System.exit(0);

9 }

10 catch(Exception e)

11 {

12 System.out.println("C");

13 }

14 finally

15 {

16 System.out.println("D");

17 }

18 }

19 }

(a) Program compiles correctly and prints "A" when executed.

(b) Program compiles correctly and prints "A" and "B" when executed.

(c) Program compiles correctly and prints "A" and "C" when executed.

(d) Program compiles correctly and prints "A","B" and "C" when executed.

Ans:

88. What will happen when you attempt to compile and run this program? [7]

public class Outer {

public String name = "Outer";

public static void main(String argv[]) {

Inner i = new Inner();

i.showName();

}//End of main

private class Inner {

String name =new String("Inner");

void showName() {

System.out.println(name);

}

}//End of Inner class

}

(a) Compile and run with output of "Outer".

(b) Compile and run with output of "Inner".

(c) Compile time error because Inner is declared as private.

(d) Compile time error because of the line creating the instance of Inner.

Ans:

89.Which of the following statements are true?

(a) An inner class may be defined as static.

(b) There are NO circumstances where an inner class may be defined as private.

(c) An anonymous class may have only one constructor.

(d) An inner class may extend another class.

Ans:

7.3) Given the following class definition, which of the following statements would be legal after the comment //Here? [7]

class InOut{

String s= new String("Between");

public void amethod(final int iArgs)

int iam;

class Bicycle{

public void sayHello(){

//Here

}//End of bicycle class

}

}//End of amethod

public void another(){

int iOther;

}

}

(a) System.out.println(s);

(b) System.out.println(iOther);

(c) System.out.println(iam);

(d) System.out.println(iArgs);

Ans:

90. Which of the following statements are true?

(a) Adding more classes via import statements will cause a performance overhead, only import classes you actually use.

(b) Under no circumstances can a class be defined with the private modifier.

(c) A inner class may under some circumstances be defined with the protected modifier.

(d) An interface cannot be instantiated.

91. The following is an outline of code for top-level class. Assume that both classes have correct constructors taking no parameters.

class NormalClass {

static class NestedClass {

// methods and variables of NestedClass

}

// methods and variables of NestedClass

}

Which of the following code fragments shows the correct way to declare and initialize a reference to a NestedClass object?

(a) NormalClass.NestedClass myNC = new NormalClass.NestedClass();

(b) NestedClass myNC = new NormalClass().new NestedClass();

(c) NestedClass myNC = new NormalClass.NestedClass();

Ans:

92.The following is an outline of code for top-level class. Assume that both classes have correct constructors taking no parameters. [3]

class BaseClass {

static class NestedClass {

// methods and variables of NestedClass

}

// methods and variables of BaseClass

}

Which of the following code fragments shows the correct way to declare and initialize a reference to a NestedClass object?

(a) BaseClass.NestedClass myNC = new BaseClass.NestedClass();

(b) NestedClass myNC = new BaseClass().new NestedClass();

(c) NestedClass myNC = new BaseClass.NestedClass();

(d) BaseClass.NestedClass nbc = new BaseClass().new NestedClass();

Ans:

93. Which of the following statements about this code are true? [4]

public class Morecombe {

public static void main(String argv[]){

Morecombe m = new Morecombe();

m.go(new Turing(){});

}

public void go(Turing t){

t.start();

}

}

class Turing extends Thread {

public void run(){

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

System.out.println(i);

}

}

}

(a) Compilation error due to malformed parameter to go method.

(b) Compilation error, class Turing has no start method.

(c) Compilation and output of 0 followed by 1.

Compilation but runtime error.

String s4 = new String(“YES”);

if ( s3 == s4 ) System.out.println(“s3 eq s4”);

}

}

(a) “equal” is printed, “s3 eq s4” is printed.

(b) “equal” is printed only.

(c) “s3 eq s4” is printed only.

(d) Nothing is printed.

Answers: b