1. Which of the following statements about abstract methods/classes in Java are FALSE?
2. Constructors cannot be abstract
3. An abstract class cannot be instantiated
4. Static methods cannot be declared abstract
5. A subclass of an abstract class must define the abstract methods

Answer: d

1. Which of these interfaces are implicitly implemented by arrays?
2. Collection
3. Cloneable
4. Iterable
5. Serializable

Answer: b, d

1. Which of the following modificators are implicitly set to any interface method?
2. abstract
3. final
4. static
5. public

Answer: a, d

1. Which of the following modificators are implicitly set to any interface field?
2. abstract
3. final
4. static
5. public

Answer: b, c, d

1. What is the output of the following code?

class SuperClass {

public void Method() {

System.out.print("SuperClass");

}

}

interface Interface {

public void Method();

}

class ChildClass extends SuperClass implements Interface {

public void Method() {

System.out.print("ChildClass");

}

}

class Test {

public static void main(String[] args) {

new ChildClass().Method();

((SuperClass)new ChildClass()).Method();

((Interface)new ChildClass()).Method();

}

}

1. ChildClassChildClassChildClass
2. ChildClasSuperClassChildClass
3. SuperClassSuperClassChildClass
4. SuperClassChildClassSuperClass

Answer: a

1. Which of the following is/are legal method declaration(s) in abstract class?
2. public abstract void print();
3. public void print();
4. final abstract print();
5. protected abstract void print();
6. private abstract void print();
7. abstract void print();

Answer: a, d, f

1. What is the output of the following code?
2. abstract class ExmplClass {
3. public void doTheThing() {
4. System.out.println("Cool!");
5. }
6. }
7. class Test {
8. public static void main(String[] args) {
9. (new ExmplClass()).doTheThing();
10. }
11. }
12. Compilation error on line 1
13. Compilation error on line 8
14. Outputs string “Cool!”
15. InstantiationException on line 8

Answer: b

1. Which is true?
2. “X extends Y” is correct if and only if X is a class and Y is an interface
3. “X extends Y” is correct if and only if X is an interface and Y is a class
4. “X extends Y” is correct if X and Y are either both classes or both interfaces
5. “X extends Y” is correct for all combinations of X and Y being classes and/or interfaces

Answer: c

1. Pick the INCORRECT statement
2. A class can implement more than one interface
3. A class can extend more than one class
4. An interface can extend more than one interface
5. An interface cannot extend a class

Answer: b

1. A class implements an interface but it does not override all the methods of that interface, then \_\_\_\_\_\_? (in JDK 7)
2. It should be declared as final class
3. It should be declare as abstract class
4. It will successfully compile
5. None of above

(What should be the answer for JDK 8?)

Answer: b

1. Which of the following variables is incorrectly declared?

interface Interface {

int a = 0;

public int b = 1;

public static int c = 2;

public final int d = 3;

public static final int e = 4;

}

1. a
2. b
3. c
4. d
5. e
6. all correct

Answer: f

1. Is the following interface valid?

public interface MyWonderfulInterface {}

1. Yes
2. No

Answer: a

13) Which of the following statements is true?

a). A class can extend multiple base classes.

b). You can implement only one interface since java does not support multiple inheritance.

c). You can implement multiple interfaces.

d). You can either extend a class or implement an interface (but not both) at a time.

Answer: C

14\*) Consider the following three classes: University, Department, and CS\_Department. The University and Department classes are related with relation r1, and the Department and CS\_Department classes are related with relation r2. Which combination of these relations is appropriate?

a). r1: inheritance, r2: inheritance

b). r1: composition, r2: inheritance

c). r1: inheritance, r2: composition

d). r1:composition,r2:composition

Answer: b. (a university has many departments, so they share a has-a relationship between them, a composition relationship. CS\_Department is a department, so these two share a is-a relationship between them, an inheritance relationship.)

15\*). You need to model a file system where there could be subfolders and files in a folder. What is the most appropriate design choice in this case to represent the relationship between Folder and File classes?

a). Use composition to model the relationship of “a Folder object consists of File objects.”

b). Use composition to model the relationship of “a Folder object consists of File objects or Folder objects.”

c). Use inheritance to define a superclass (say FolderItem) and make Folder and File classes subclasses to this class. Use composition to model the relationship “a Folder object consists of FolderItem objects.”

d). Use inheritance between Folder and File classes to model the relationship “a Folder is of type File.”

Answer: C. (in fact, this arrangement is referred to as a composite design pattern)

16) Pick the correct statement(s)

a). abstract class can declare and define constructor

b). abstract class can be final

c). abstract class can declare and define static methods

d). it is necessary for abstract class to have abstract method

e). abstract class can contain main method

Answer: A, C and E

1. Since you cannot create instance of abstract class, constructor can only be called during constructor chaining, i.e. when you create instance of concrete implementation class.
2. You must follow guidelines for making a method static in Java, as it’s not welcomed in a object oriented design, because static methods can not be overridden in Java.

17) Given:

interface Machine { }

interface Engine { }

abstract interface Tractor extends Machine, Engine {

void pullStuff();

}

class Deere implements Tractor {

public void pullStuff() { System.out.print("pulling ");}

}

class LT255 implements Tractor extends Deere {

public void pullStuff() {

System.out.print("pulling harder ");

}

}

public class LT155 extends Deere implements Tractor, Engine { }

What is the result? (Choose all that apply.)

A. Compilation succeeds.

B. Compilation fails because of error(s) in Tractor.

C. Compilation fails because of error(s) in Deere.

D. Compilation fails because of error(s) in LT255.

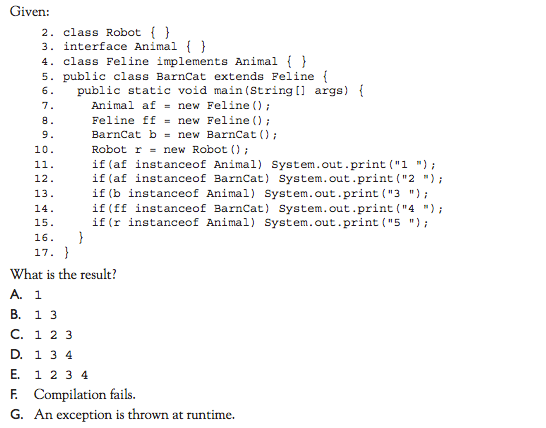
E. Compilation fails because of error(s) in LT155.

Answer: D

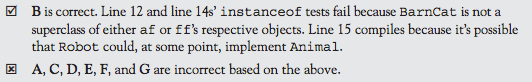
D is correct. When a class implements and extends, the extends declaration comes first.

A, B, C, and E are incorrect because all of the other code is legal. It is legal to “re-extend” an interface, and it’s legal to implement several interfaces.

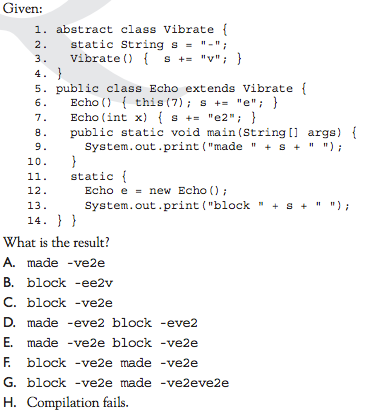
18)

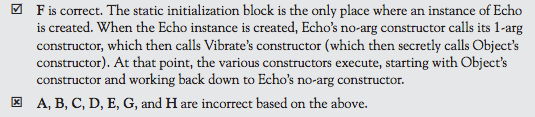


(Class Robot is defined somewhere)

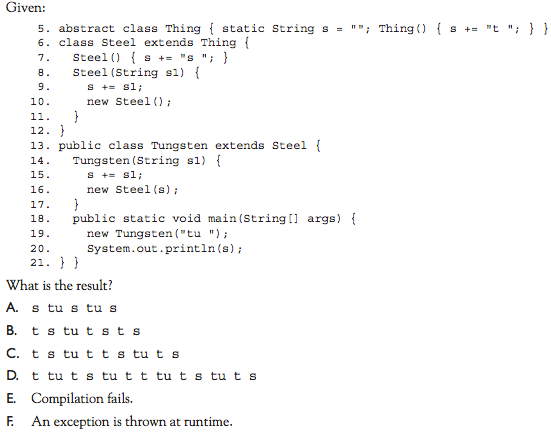


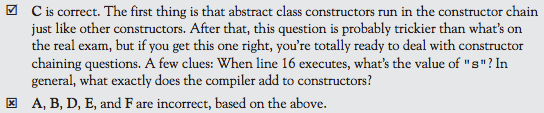
19)





20)





21)

