

Declarations and Access Control

Author: JavaChamp Team

Senior Java Developer @QuizOver.com

Copyright (c) 2014-2015

Create, Share, and Discover Online Quizzes.

QuizOver.com is an intuitive and powerful online quiz creator. [learn more](#)

Join QuizOver.com



How to Analyze Stocks

By Yasser Ibrahim

1 month ago
12 Responses

© iStock: Thomson Moter



Pre Employment English

By Katharina jennifer N

5 months ago
19 Responses

© iStock: Albin



Lean Startup Quiz

By Yasser Ibrahim

2 months ago
16 Responses

© iStock: Gekwini Okun

Powered by QuizOver.com

The Leading Online Quiz & Exam Creator

Create, Share and Discover Quizzes & Exams

<http://www.quizover.com>

Disclaimer

All services and content of QuizOver.com are provided under QuizOver.com terms of use on an "as is" basis, without warranty of any kind, either expressed or implied, including, without limitation, warranties that the provided services and content are free of defects, merchantable, fit for a particular purpose or non-infringing.

The entire risk as to the quality and performance of the provided services and content is with you.

In no event shall QuizOver.com be liable for any damages whatsoever arising out of or in connection with the use or performance of the services.

Should any provided services and content prove defective in any respect, you (not the initial developer, author or any other contributor) assume the cost of any necessary servicing, repair or correction.

This disclaimer of warranty constitutes an essential part of these "terms of use".

No use of any services and content of QuizOver.com is authorized hereunder except under this disclaimer.

The detailed and up to date "terms of use" of QuizOver.com can be found under:

<http://www.QuizOver.com/public/termsOfUse.xhtml>

eBook Content License

Creative Commons License

Attribution-NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0)

<http://creativecommons.org/licenses/by-nc-nd/3.0/>

You are free to:

Share: copy and redistribute the material in any medium or format

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution: You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial: You may not use the material for commercial purposes.

NoDerivatives: If you remix, transform, or build upon the material, you may not distribute the modified material.

No additional restrictions: You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Table of Contents

Quiz Permalink: <http://www.quizover.com/question/java-declarations-and-access-control-certification-questions>

Author Profile: <http://www.quizover.com/user/profile/Java.Champ>

1. Declarations and Access Control

4. Chapter: Declarations and Access Control

1. Declarations and Access Control Questions

4.1.1. What is the result of compiling and running the following code?

Author: Yasser Ibrahim

What is the result of compiling and running the following code?

```
public class Tester {  
    static int x = 4;  
  
    public Tester() {  
        System.out.print(this.x); // line 4  
        Tester();  
    }  
  
    public static void Tester() { // line 8  
        System.out.print(this.x); // line 9  
    }  
  
    public static void main(String... args) { // line 12  
        new Tester();  
    }  
}
```

Please choose only one answer:

- Compile error at line 4 (static x must be only accessed inside static methods)
- Compile error at line 8 (constructors can't be static)
- Compile error at line 9 (static methods can't invoke this)
- Compile error at line 12 (invalid argument type for method main)
- 44

Check the answer of this question online at QuizOver.com:

Question: [Can java static methods invoke this?](#)

Flashcards:

<http://www.quizover.com/flashcards/can-java-static-methods-invoke-this?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/can-java-static-methods-invoke-this?pdf=3044>

4.1.2. What is the result of compiling and running the following code?

Author: Yasser Ibrahim

What is the result of compiling and running the following code?

```
public class Tester {  
  
    static int x = 4;  
    int y = 9;  
  
    public Tester() {  
        System.out.print(this.x); // line 7  
        printVariables();  
    }  
  
    public static void printVariables() {  
        System.out.print(x); // line 12  
        System.out.print(y); // line 13  
    }  
  
    public static void main(String... args) { // line 16  
        new Tester();  
    }  
}
```

Please choose only one answer:

- Compile error at line 7 (static x must be only accessed inside static methods)
- Compile error at line 13 (static methods cant make reference to non-static variables)
- Compile error at line 16 (invalid argument type for method main)
- 49
- Compile error at line 12 (must access x by writing Tester.x)

Check the answer of this question online at QuizOver.com:

Question: [What can java static methods make references to?](http://www.quizover.com/question/what-can-java-static-methods-make-references-to?pdf=3044)

Flashcards:

<http://www.quizover.com/flashcards/what-can-java-static-methods-make-references-to?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/what-can-java-static-methods-make-references-to?pdf=3044>

4.1.3. You have two packages, trunk1 and trunk2 where class Sheet declared...

Author: Yasser Ibrahim

You have two packages, trunk1 and trunk2 where class Sheet declared in trunk1

and class Container declared in trunk2, the following code contains a compilation error, where could it be?

```
package trunk1;

public class Sheet {

    public static int pageNumber = 99; // line 1
    Sheet() {} // line 2
}

package trunk2;

import trunk1.Sheet;

public class Container {

    public static void main(String... args) { //line 1
        System.out.print(Sheet.pageNumber); //line 2
        Sheet sheet = new Sheet(); //line 3
    }
}
```

Please choose only one answer:

- In package trunk1 at line 2, constructor Sheet must be marked public like its class
- In package trunk2 at line 1, invalid string argument for method main()
- In package trunk2 at line 2, invalid access to member pageNumber
- In package trunk2 at line 3, invalid call to constructor Sheet()

Check the answer of this question online at QuizOver.com:

Question: [How to access a class from another package in java?](http://www.quizover.com/question/how-to-access-a-class-from-another-package-in-java?pdf=3044)

Flashcards:

<http://www.quizover.com/flashcards/how-to-access-a-class-from-another-package-in-java?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/how-to-access-a-class-from-another-package-in-java?pdf=3044>

4.1.4. You have three packages, trunk1, trunk2 and trunk3, What is the expected output of compiling and running file Tester?

Author: JavaChamp Team

You have three packages, trunk1, trunk2 and trunk3, What is the expected output of compiling and running file Tester?

```
package trunk1;

public class Account {

    Account() { //line 1
        System.out.println("Account");
    }
}

package trunk2;

import trunk1.Account;

public class CurrentAccount extends Account{
    public CurrentAccount() { // line 2
        System.out.println("CurrentAccount");
    }
}

package trunk3;

import trunk1.Account;
import trunk2.CurrentAccount; // line 3

public class Tester {

    public static void main(String[] args) {
        Account c = new Account(); // line 4
        CurrentAccount ca = new CurrentAccount();
    }
}
```

Please choose all the answers that apply:

- Account
CurrentAccount
- Account
Account
CurrentAccount
- Compilation error in package trunk1 at line 1
- Compilation error, in package trunk2 at line 2

- Compilation error, in package trunk3 at line 3
- Compilation error, in package trunk3 at line 4

Check the answer of this question online at QuizOver.com:

Question: [packages and access scope in java](#)

Flashcards:

<http://www.quizover.com/flashcards/packages-and-access-scope-in-java?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/packages-and-access-scope-in-java?pdf=3044>

4.1.5. What is the result of compiling and running the following code?

Author: Yasser Ibrahim

What is the result of compiling and running the following code?

```
public class Tester {  
    static int p = test(); //line 1  
  
    static public int test() {  
        System.out.print(p); //line 4  
        return 99;  
    }  
  
    public static void main(String[] args)  
    {  
        System.out.print(p); //line 11  
    }  
}
```

Please choose only one answer:

- 099
- Compilation error at line 1, p must be initialized by a value
- Compilation error at line 4, using uninitialized variable p
- Compilation error at line 11, p must be called using its class by writing Tester.p

Check the answer of this question online at QuizOver.com:

Question: [How to use java static members?](#)

Flashcards:

<http://www.quizover.com/flashcards/how-to-use-java-static-members?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/how-to-use-java-static-members?pdf=3044>

4.1.6. Consider the following two classes declared and defined in two diff...

Author: JavaChamp Team

Consider the following two classes declared and defined in two different packages, what can be added in class B to form what considered a correct access to class A from main() method of class B?

```
package subPackage;

public class A { }

package anotherPackage;
// line 1
public class B {
public static void main(String[] args) {
// line 2
}
}
```

Please choose all the answers that apply:

- at line 1 add nothing
at line 2 add : new A();
- at line 1 add: import package.*;
at line 2 add : new subPackage.A();
- at line 1 add: import subPackage.*;
at line 2 add : new A();
- at line 1 add: import subPackage.A;
at line 2 add : new A();
- at line 1 add nothing
at line 2 add : new subPackage.A();

Check the answer of this question online at QuizOver.com:

Question: [How to access java class from another package?](#)

Flashcards:

<http://www.quizover.com/flashcards/how-to-access-java-class-from-another-package?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/how-to-access-java-class-from-another-package?pdf=3044>

4.1.7. Given the following declarations, what is considered a correct stat...

Author: JavaChamp Team

Given the following declarations, what is considered a correct statement?

```
interface Chewable {}  
interface Eatable extends Chewable{}  
class Vitamin{}  
class Food implements Eatable { Vitamin[] vitamins;}
```

Please choose all the answers that apply:

- Eatable is Chewable
- Vitamin is Eatable
- Food is Chewable
- Food has Vitamin
- Vitamin has Food
- Food is an Object

Check the answer of this question online at QuizOver.com:

Question: [types of relationships between java classes](#)

Flashcards:

<http://www.quizover.com/flashcards/types-of-relationships-between-java-classes?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/types-of-relationships-between-java-classes?pdf=3044>

4.1.8. What is the result of compiling and running the following code?

Author: JavaChamp Team

What is the result of compiling and running the following code?

```
class Base {  
    private Base() {System.out.print("Base");}  
}  
  
public class Derived extends Base {  
    public Derived() {System.out.print("Derived");}  
  
    public static void main(String[] args) {  
        new Derived();  
    }  
}
```

Please choose only one answer:

- BaseDerived
- Derived
- Exception is thrown at runtime
- Compilation error

Check the answer of this question online at QuizOver.com:

Question: [Can java constructor be private?](#)

Flashcards:

<http://www.quizover.com/flashcards/can-java-constructor-be-private?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/can-java-constructor-be-private?pdf=3044>

4.1.9. Which of the following is true?

Author: Yasser Ibrahim

Which of the following is true?

Please choose all the answers that apply:

- When a class marked final, it cannot be extended
- When a method marked final, it cannot be overridden
- When a method marked final, it cannot be overloaded
- Local variables of a method cannot be final
- A variable of a primitive type (int,float,..) which marked final, its value at initialization cannot be changed

Check the answer of this question online at QuizOver.com:

Question: [what does final do in java?](#)

Flashcards:

<http://www.quizover.com/flashcards/what-does-final-do-in-java?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/what-does-final-do-in-java?pdf=3044>

4.1.10. What does it mean to mark a class strictfp?

Author: JavaChamp Team

What does it mean to mark a class strictfp?

Please choose only one answer:

- it means this class can't be subclassed
- it means this class can never be instantiated and it is to be extended
- it means that any method code in the class will conform to the IEEE 754 standard rules for floating points

Check the answer of this question online at QuizOver.com:

Question: [strictfp in java](#)

Flashcards:

<http://www.quizover.com/flashcards/strictfp-in-java?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/strictfp-in-java?pdf=3044>

4.1.11. What can be marked with strictfp modifier?

Author: Yasser Ibrahim

What can be marked with strictfp modifier?

Please choose all the answers that apply:

- an interface
- a class
- a member method
- a member field
- a local variable

Check the answer of this question online at QuizOver.com:

Question: [strictfp modifier in java](#)

Flashcards:

<http://www.quizover.com/flashcards/strictfp-modifier-in-java?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/strictfp-modifier-in-java?pdf=3044>

4.1.12. A top level Java class can have the following modifiers:

Author: JavaChamp Team

A top level Java class can have the following modifiers:

Please choose all the answers that apply:

- strictfp and final
- abstract and final
- public and abstract
- protected
- private

Check the answer of this question online at QuizOver.com:

Question: [top level class modifiers in java](#)

Flashcards:

<http://www.quizover.com/flashcards/top-level-class-modifiers-in-java?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/top-level-class-modifiers-in-java?pdf=3044>

4.1.13. An inner class in java can have the following modifiers (at the sam...

Author: JavaChamp Team

An inner class in java can have the following modifiers (at the same time):

Please choose all the answers that apply:

- public and abstract
- abstract and final
- protected
- private

Check the answer of this question online at QuizOver.com:

Question: [inner class modifiers in java](#)

Flashcards:

<http://www.quizover.com/flashcards/inner-class-modifiers-in-java?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/inner-class-modifiers-in-java?pdf=3044>

4.1.14. What is the output ?

Author: Nikhil Kumar

What is the output ?

```
class InitDemo{
    static int i=demo();
    static{System.out.println(i);}
    InitDemo(){
        System.out.println("hello 1");
    }
    public static void main(String... args){
        System.out.println("Hello 2");
    }
    static int demo(){
        System.out.println("Inside Demo");
        return 10;
    }
}
```

Please choose only one answer:

- Compilation error
- IllegalArgumentException is thrown at runtime.
- Inside Demo
10
Hello 2
- Hello 2
Inside Demo
10

Check the answer of this question online at QuizOver.com:

Question: [static public private access modifiers java nikhil kumar](http://www.quizover.com/question/static-public-private-access-modifiers-java-nikhil-kumar?pdf=3044)

Flashcards:

<http://www.quizover.com/flashcards/static-public-private-access-modifiers-java-nikhil-kumar?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/static-public-private-access-modifiers-java-nikhil-kumar?pdf=3044>

4.1.15. This is legal or not?

Author: MrDick

This is legal or not?

```
final strictfp public class Parent implements Serializable {}
```

Please choose only one answer:

- legal and compile free
- No, there is compilation error(s)

Check the answer of this question online at QuizOver.com:

Question: [java final strictfp public class](#)

Flashcards:

<http://www.quizover.com/flashcards/java-final-strictfp-public-class?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/java-final-strictfp-public-class?pdf=3044>

4.1.16. Which of the following are NOT LEGAL Java identifiers according to ...

Author: Alinaloana Florea

Which of the following are NOT LEGAL Java identifiers according to the "java language specification"?

```
int ; // line 1
int _____9_y; // line 2
int €a; // line 3
int £b_, c; // line 4
int this_is_a_way_too_long_name_for_an_identifier; // line 5
int d3#; // line 6
int -e, 7g, :h; // line 7
float .f; // line 8
double _$$_; // line 9
int true = 10; // line 10
int null = 4; // line 11
int foo = 3; // line 12
int Foo = 3; // line 13
int continue = 6; // line 14
int Volatile = 6; // line 15
public abstract void addEvent(); // line 16
```

Please choose all the answers that apply:

- lines 1, 3, 4 - contain invalid characters for Java identifiers
- line 5 - is too long
- line 16 - is not a valid JavaBean identifier for a method that wants to add a listener to an event source
- lines 6 and 8 - can NOT contain "#" and "." even if assigned to a numeric or floating point variable
- line 7 - can NOT contain the characters "-" and ":", nor can a Java identifier start with a digit
- lines 10, 11 and 14

Check the answer of this question online at QuizOver.com:

Question: [Alinaloana Florea legal java identifiers](http://www.quizover.com/question/question-alinaloana-florea-legal-java-identifiers?pdf=3044)

Flashcards:

<http://www.quizover.com/flashcards/question-alinaloana-florea-legal-java-identifiers?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/question-alinaloana-florea-legal-java-identifiers?pdf=3044>

4.1.17. Which of the following are NOT valid JavaBeans identifiers for meth...

Author: Alinaloana Florea

Which of the following are NOT valid JavaBeans identifiers for methods that add/remove listeners from an event source?

```
// Event subscribers:  
public void addXListener(MyListener m) // line 1  
void addXListener() // line 2  
public void addEvent(Event e) // line 3  
public void addEventListener(EventListener e) // line 4  
// Event removers:  
public void removeMyListener(MyListener m) // line 5  
public void removeMyListener() // line 6  
public void deleteActionListner(ActionListner m) // line 7
```

Please choose all the answers that apply:

- lines 1 and 3 - the names of the methods do not match the type of argument they receive + JavaBeans adder/remover methods should always end with "Listener"
- line 2 - JavaBeans methods must be public in order to provide access to the private variables of a class + it's missing an argument of type XListener
- line 4 - it's adding or missing something
- line 5 - the name of its argument should be more meaningful
- line 6 - it misses the event to be removed
- line 7 - should be "remove" instead of "delete"

Check the answer of this question online at QuizOver.com:

Question: [illegal javabeans identifiers for methods that add /remove listeners](#)

Flashcards:

<http://www.quizover.com/flashcards/illegal-javabeans-identifiers-for-methods-that-add-remove-listeners?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/illegal-javabeans-identifiers-for-methods-that-add-remove-listeners?pdf=3044>

4.1.18. Which of the following are NOT valid JavaBeans method identifiers f...

Author: Alinaloana Florea

Which of the following are NOT valid JavaBeans method identifiers for setters/getters of private class variables?

```
// Property setters:
void setCustomername() // line 1
public int modifyValue(int val) // line 2
public void setCustomerName(String customer) // line 3
public void setIsAlive(boolean isAlive) // line 4
public void addSize(int size) // line 5
public void putSize(int size) // line 6
// Property getters:
public int getMyValue() // line 7
public int getAccount#(String clientName) // line 8
protected boolean getHappy() // line 9
public boolean isHappy() // line 10
public boolean getHappy() // line 11
public int isNrOfFlowers() // line 12
```

Please choose all the answers that apply:

- lines 1 and 9 - JavaBeans methods must be public in order to provide access to the private variables of a class
- lines 2, 5 and 6 - according to JavaBeans "add" should be used only with listener methods, and "set" should be used instead of "put" and "modify"
line 8 - will produce a compile-time error due to invalid Java identifier
- line 3 - the method's argument name should match the one its name: "customerName"
- lines 4, 7 and 11 - are adding or missing some things
- line 10 - it should be getHappy() because it's a getter method as it returns a value
- line 12 - "nrOfFlowers" seems to be of type "int" not "boolean", as that's what the method is returning, so it should be "get" instead of "is"

Check the answer of this question online at QuizOver.com:

Question: [legal javabeans method identifiers for setters /getters](#)

Flashcards:

<http://www.quizover.com/flashcards/legal-javabeans-method-identifiers-for-setters-getters?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/legal-javabeans-method-identifiers-for-setters-getters?pdf=3044>

4.1.19. Correct the code in AnimalXyZ.java:

Author: Alinaloana Florea

Correct the code in AnimalXyZ.java:

```
public /*x1*/ class /*x2*/ Animal implements Runnable {}

package test2;
/*x3*/import java.io.Serializable;

/*x4*/public class Dog extends Animal implements /*x5*/ Serializable
{
    void /*x6*/ doStuff(/*x7*/int arg1, int /*x8*/ arg2) {} // wrong comments
}

class X{}
class Y{}
```

Please choose all the answers that apply:

- remove the "x#" comments, they are placed in wrong places
- move the package and the import statements at the beginning of the file
- make "Animal" and "Dog" abstract as they do NOT implement the run() from the Runnable interface
- remove the "public" from "Dog", as only one class can be public in a .java file
- - rename "Animal" to "AnimalXyZ" as the only public class in a .java file must match the name of the file
- OR make ALL classes in the file "default"
- "import java.lang.Runnable;" is necessary at beginning of the file

Check the answer of this question online at QuizOver.com:

Question: [java comments packages imports](#)

Flashcards:

<http://www.quizover.com/flashcards/java-comments-packages-imports?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/java-comments-packages-imports?pdf=3044>

4.1.20. Choose all the lines which if inserted independently instead of "//...

Author: Alinaloana Florea

Choose all the lines which if inserted independently instead of "//insert code here" will allow the following code to compile:

```
public class Test
{
    public static void main(String[] args)
    {
        add();
        add(1);
        add(1, 2);
    }

    // insert code here
}
```

Please choose only one answer:

- 1. void add(Integer... args){}
- 2. static void add(int... args, int y){}
- 3. static void add(int args...){}
- 4. static void add(int[]... args){}
- 5. static void add(int...args){} ;

Check the answer of this question online at [QuizOver.com](http://www.quizover.com):

Question: [correct usage of var-args Choose all the Alinaloana Oracle Certified](#)

Flashcards:

<http://www.quizover.com/flashcards/question-correct-usage-of-var-args-choose-all-the-alinaioana-oracle-ce?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/question-correct-usage-of-var-args-choose-all-the-alinaioana-oracle-ce?pdf=3044>

4.1.21. Correct the following code: (consider the packages as each being in...

Author: Alinaloana Florea

Correct the following code: (consider the packages as each being in a different file)

```
package pkgA;

public class X1
{
    X1() // line 1
    {
        public int x1 = 11; // line 2
        protected int x2=22; // line 2
        int x3 = 33;
        private int x4 = 44; // line 2

        System.out.println(x1 + "" + x2 + x3 + x4); // line 3
    }

    public int x1 = 1;
    static protected int x2 = 2;
    int x3 = 3;
    private int x4 = 4;
}

class X2 extends X1
{
    private int x4 = 4; // line 3.5

    public X2()
    {
        System.out.println(x1 + x2 + x3 + x4); // line 4

        X1 x1 = new X1();
        System.out.println(x1.x1 + x1.x2 + x1.x3 + x1.x4); // line 4
    }
}

protected class X3 {} // line 5
private class X4 {} // line 5

//
package pkgB;
import pkgA.*;

class Y extends X1
{
    Y()
    {
        System.out.println(x1 + x2 + x3 + x4); // line 6.1

        X2 x2 = new X2(); // line 7

        X1 x1 = new X1(); // line 8
        System.out.println(x1.x1+ x1.x2 + x1.x3 + x1.x4); // line 6.2
    }
}

public class Test extends Y
{
```

```
public static void main(String[] args)
{
    new Test().go();
}

void go()
{
    System.out.println(x1 + x2); // line 6.1

    Y y = new Y();
    System.out.println(y.x1+ y.x2 + y.x3 + y.x4); // line 6.2
}
}
```

Please choose all the answers that apply:

- line 1 - X1's constructor must be public for "new" to succeed on line 8. And a similar thing is on line 7, which will work only if the class X2 is made public in its package.
- lines 2 - local/method variables can be only "final" or "default"
- line 3 - won't compile because of the quotes added to those integers
line 3.5 - is not allowed to define a var with the same name as in the superclass
- lines 4 - can not print "x4" as it's private in class X1 so it can be used only inside X1
- lines 5 - when defined in a package, classes are allowed to be also static, private or protected
- lines 6 - should follow the same rules as the lines 4: "default" and "private" variables must be removed + the "protected" vars kept only when used through inheritance: on lines 6.1

Check the answer of this question online at QuizOver.com:

Question: [what are java access modifiers?](#)

Flashcards:

<http://www.quizover.com/flashcards/what-are-java-access-modifiers?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/what-are-java-access-modifiers?pdf=3044>

4.1.22. (SCJP6) Correct the following code:

Author: Alinaloana Florea

(SCJP6) Correct the following code:

```
final class X1
{
    final X1() { System.out.print(B); } // line 1

    final int B = 9;

    final private void doStuff() {} // line 2
}

class X2 extends X1 // line 3
{
    final int B = 9;

    final static void doStuff() { System.out.println("hello from X2"); }
}

public class Test extends X2
{
    public static void main(String[] args)
    {
        doStuff();
        new Test().doStuff(3);
    }

    static void doStuff() {} // line 4
    void doStuff(int arg) { System.out.println("hello from Test"); } // line 5
}
```

Please choose all the answers that apply:

- line 1 - can not make a constructor final because it's not inheritable so it can not be anyway overridden
- lines 3 - can not extend/subclass a final class
- Can not use in class X2 the same variable and method names as in its superclass X1
- line 4 - can not override a final inherited method
- line 5 - can not overload a final inherited method
- line 2 - "final" can not be mixed with "private"

Check the answer of this question online at QuizOver.com:

Question: [what are java final primitives, classes and methods?](http://www.quizover.com/question/what-are-java-final-primitives-classes-and-methods?pdf=3044)

Flashcards:

<http://www.quizover.com/flashcards/what-are-java-final-primitives-classes-and-methods?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/what-are-java-final-primitives-classes-and-methods?pdf=3044>

4.1.23. (SCJP 6) Find the errors in:

Author: Alinaloana Florea

(SCJP 6) Find the errors in:

```
static class X // line 1
{
    static { a = 3; } // line 2

    static X() {} // line 1

    int a;
    static int b;

    void doX(){ System.out.println("In X" + a + b); } // line 3
    static void doStuff() { System.out.println("In X" + /*a*/ + b); } // line 4
}

public class Test extends X
{
    public static void main(String[] args)
    {
        X x = new X();

        doStuff();
        x.doStuff(); // line 5
        X.doStuff(); // line 5.2
    }

    void doStuff() // line 6
    {
        System.out.println("In Test");
    }
}
```

Please choose all the answers that apply:

- lines 1 - "static" can not mark classes and constructors
- line 2 - "static" can not mark an initialization block {}
- lines 2 - "b" must be used instead of "a" as in a static context(method/init block) it's possible to access only static variables
- lines 3 - a non-static context can access only non-static variables
- line 4 - a static method can access non-static variables("a"), but an error will occur because of the way "a" is commented leaving two "+" inside the print()
- line 6 - to be a valid override it must not be less accessible, so it must keep its static status + "static" is needed also because it's called directly in the static method main()

Check the answer of this question online at QuizOver.com:

Question: [what are java static non inner classes?](#)

Flashcards:

<http://www.quizover.com/flashcards/what-are-java-static-non-inner-classes?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/what-are-java-static-non-inner-classes?pdf=3044>

4.1.24. Spot the errors in:

Author: Alinaloana Florea

Spot the errors in:

```
static int x; // line 1
static { System.out.println(x); x = 1; } // line 1

class X
{
    int x;
    static int y;
    static Integer z = new Integer(1);

    System.out.println(x + y); // line 2

    X()
    {
        for(int __x = 0; __x < 3; __x++) ;
        int #lb = 7; // line 3

        long [] arr [5]; // line 4
        Boolean []b[];

        System.out.println(x + y + z); // line 5
        System.out.println(x + y + "" + z); // line 6

        System.out.println(new X()); // line 7
    }
}
```

Please choose all the answers that apply:

- lines 1 - everything in Java must be within a class
line 2- can not print here
- line 3 - will produce a compilation error because it's using an invalid Java identifier
- the "for" statement before line 3 will not compile because of the semicolon following it
- line 4 - an array must specify its size only when creating an object of its type not at declaration
- line 6 - will not compile because of the added quotes
- line 7 - will produce a run-time infinite loop which will crash JVM

Check the answer of this question online at QuizOver.com:

Question: [how to write java variable declarations?](#)

Flashcards:

<http://www.quizover.com/flashcards/how-to-write-java-variable-declarations?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/how-to-write-java-variable-declarations?pdf=3044>

4.1.25. (SCJP6) Correct the following code:

Author: Alinaloana Florea

(SCJP6) Correct the following code:

```
class X
{
    X()
    {
        System.out.print(B);

        final int B; // line 1
        B=7;
        System.out.print(B);
        B=8; // line 2
        System.out.print(B);
    }

    int a;
    final int B = 9; // line 3
}

public class Test
{
    public static void main(String[] args)
    {
        final X x = new X();
        System.out.print(x.a);

        x = new X(); // line 4
        x.a = 3; // line 5
        System.out.println(x.a);
    }
}
```

Please choose only one answer:

- Final variables must be assigned an initial value, so the final variable "B" from line 1 must receive an initial value at its declaration time
- Lines 2 and 4 must be removed as final variables can not be reassigned
- Line 5 must be removed as is not allowed to modify the object referred by a final reference variable

Check the answer of this question online at QuizOver.com:

Question: [final variables SCJP6 Correct the following Alinaloana Oracle Quest](#)

Flashcards:

<http://www.quizover.com/flashcards/question-final-variables-scp6-correct-the-following-alinaioana-oracle?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/question-final-variables-scp6-correct-the-following-alinaioana-oracle?pdf=3044>

4.1.26. (SCJP 6) Find the errors in:

Author: Alinaloana Florea

(SCJP 6) Find the errors in:

```
class X
{
    X()
    {
        System.out.print(a);

        static int a; // line 1
        a=4;

        System.out.print(a);
        System.out.print(b);
    }

    int a;
    static int b;
}

public class Test extends X
{
    public static void main(String[] args)
    {
        X x = new X();

        System.out.println(a + "" + x.a + X.a); // line 2
        System.out.println(b + "" + x.b + X.b);
    }
}
```

Please choose all the answers that apply:

- line 1 - inside a method there can be only "final" and "default" variables, so line 1 should be removed
- line 1 - it would anyway not be allowed to define another "int a;" inside the constructor X()
- line 2 - "a" is not static inside X so it can not be called directly in a static context/main(), nor can it be called directly on class X
- The print lines inside main() will produce a compile-time error due to the quotes added to integer variables
- Static variables must be given a default value at their declaration time

Check the answer of this question online at QuizOver.com:

Question: [how to write java non static variables?](http://www.quizover.com/question/how-to-write-java-non-static-variables?pdf=3044)

Flashcards:

<http://www.quizover.com/flashcards/how-to-write-java-non-static-variables?pdf=3044>

Interactive Question:

<http://www.quizover.com/question/how-to-write-java-non-static-variables?pdf=3044>