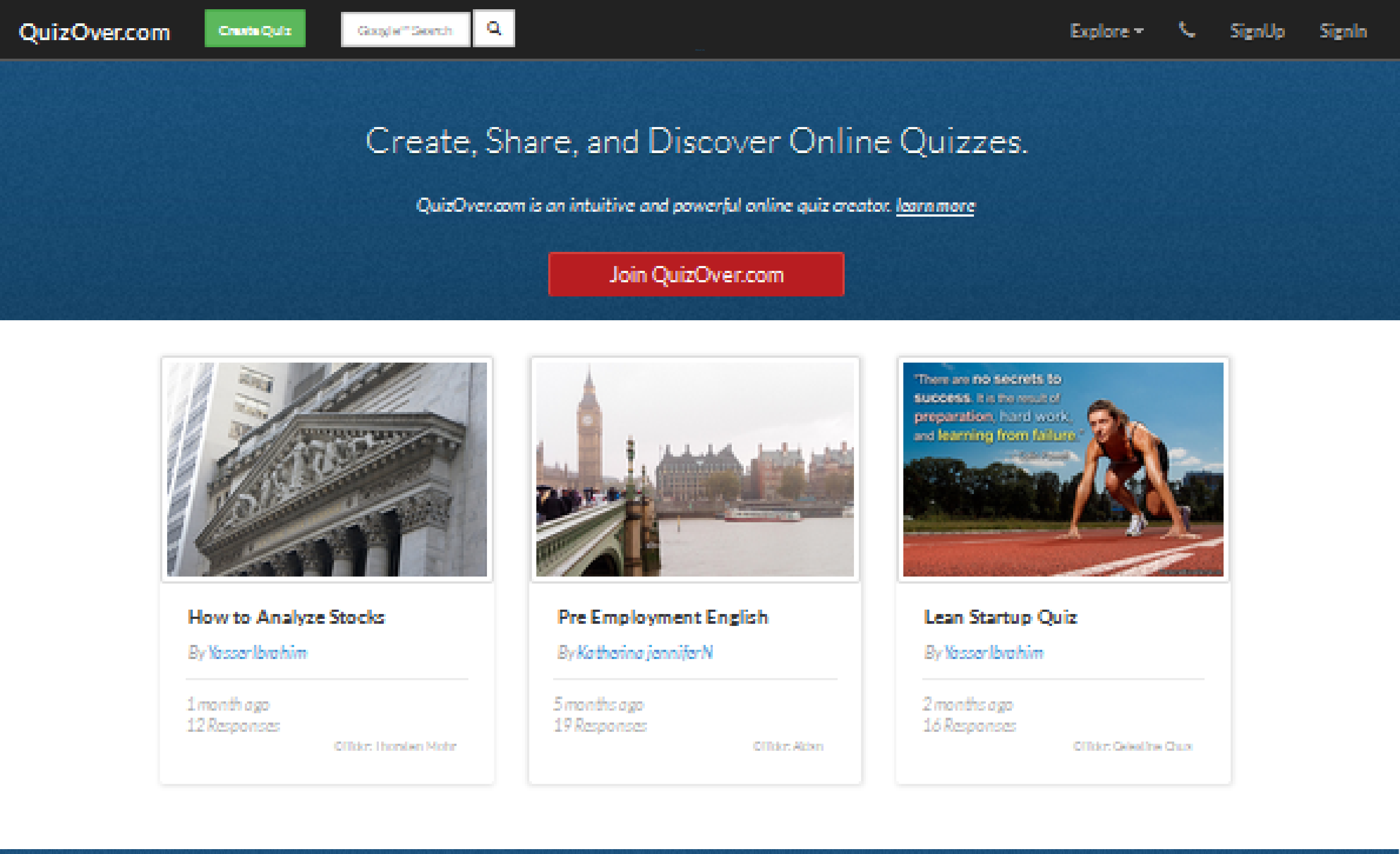
Interfaces and Abstract Classes

Author: JavaChamp Team

Senior Java Developer @QuizOver.com



Create, Share, and Discover Online Quizzes.

QuizOver.com is an intuitive and powerful online quiz creator. learn more

How to Analyze Stocks
By YasserIbrahim
1 month ago
12 Responses
©flickr: Thorsten Mohr

Pre Employment English
By Katherina jenniferN
5 months ago
19 Responses
©flickr: Aldon

Lean Startup Quiz
By YasserIbrahim
2 months ago
16 Responses
©flickr: Celestine Chua

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

Table of Contents

Quiz Permalink: http://www.quizover.com/question/java-interfaces-and-abstract-classes-certification-questions

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

4. Chapter: Interfaces and Abstract Classes

1. Interfaces and Abstract Classes Questions

4.1.1. Is the following declaration for interface Bendable correct and fre...

Author: Yasser Ibrahim

Is the following declaration for interface Bendable correct and free of compilation error?

```
abstract interface Bendable { // line 1

 final int x = 2009; // line 3

 void method1() ; // line 5
 public static class Angle {} // line 6
}
```

Please choose only one answer:

- Yes, this is a correct and free of error declaration
- No, compilation error at line 1 , abstract should be removed
- No, compilation error at line 3 , x should be declared public final
- No, compilation error at line 5 , method method1() should be declared public abstract
- No, compilation error at line 6 , can't declare a class inside an interface

Check the answer of this question online at QuizOver.com:
Question: How to declare a class inside an interface in java?

Flashcards:
http://www.quizover.com/flashcards/how-to-declare-a-class-inside-an-interface-in-java?pdf=3044

Interactive Question:
http://www.quizover.com/question/how-to-declare-a-class-inside-an-interface-in-java?pdf=3044

4.1.2. Is the following declaration for interface Bendable correct and fre...

Author: Yasser Ibrahim

Is the following declaration for interface Bendable correct and free of compilation error?

```
abstract interface Bendable { // line 1

 final int x = 2009; // line 3

 void method1(); // line 5

}
```

Please choose only one answer:

- Yes, this is a correct and free of error declaration
- No, compilation error at line 1, Bendable should be declared public abstract
- No, compilation error at line 3 , x should be declared public final
- No, compilation error at line 5 , method method1() should be declared public abstract

Check the answer of this question online at QuizOver.com:
Question: How to declare java interface?

Flashcards:
http://www.quizover.com/flashcards/how-to-declare-java-interface?pdf=3044

Interactive Question:
http://www.quizover.com/question/how-to-declare-java-interface?pdf=3044

4.1.3. Is the following declaration for interface Bendable correct and fre...

Author: JavaChamp Team

Is the following declaration for interface Bendable correct and free of compilation error?

```
abstract interface Bendable { // line 1

 final int x = 2009; // line 3

 void method1(){}; // line 5

}
```

Please choose only one answer:

- Yes, this is a correct and free of error declaration
- No, compilation error at line 1
- No, compilation error at line 3
- No, compilation error at line 5

Check the answer of this question online at QuizOver.com:
Question: What are java interface legal declaration rules?

Flashcards:
http://www.quizover.com/flashcards/what-are-java-interface-legal-declaration-rules?pdf=3044

Interactive Question:
http://www.quizover.com/question/what-are-java-interface-legal-declaration-rules?pdf=3044

4.1.4. Will the following code compile correctly?

Author: Yasser Ibrahim

Will the following code compile correctly?

```
abstract class AirPlane {

 abstract void fly();

 void land() { // line 5
  System.out.print("Landing..");
 }

}

class AirJet extends AirPlane {

 AirJet() {
  super(); // line 14
 }

 void fly() {
  System.out.print("Flying..");
 }
}
```

Please choose only one answer:

- Yes, it will compile with no errors
- No, because at line 5 method land() must be abstract since class AirPlane is abstract
- No, because class AirJet must override method land()
- No, because at line 14 AirJet constructor is calling the super() while AirPlane has no constructor defined

Check the answer of this question online at QuizOver.com:
Question: How to extend java abstract class?

Flashcards:
http://www.quizover.com/flashcards/how-to-extend-java-abstract-class?pdf=3044

Interactive Question:
http://www.quizover.com/question/how-to-extend-java-abstract-class?pdf=3044

4.1.5. The following code contains a compilation error , what can be done ...

Author: Yasser Ibrahim

The following code contains a compilation error , what can be done to fix this error - independently?

```
abstract class AirPlane { // line 1
 abstract void fly(); // line 2

 void land() {
  System.out.print("Landing..");
 }

}

class AirJet extends AirPlane { // line 10

 AirJet() {
  super(); // line 13
 }

 void fly() {
  System.out.print("Flying..");
 }

 abstract void land() ; // line 20
}
```

Please choose all the answers that apply:

- Remove abstract from line 20 and add body to method land()
- Declare class AirJet as abstract to at line 10
- Remove super() call at line 13
- Remove abstract at line 1 and line 2

Check the answer of this question online at QuizOver.com:
Question: Extending java abstract class

Flashcards:
http://www.quizover.com/flashcards/extending-java-abstract-class?pdf=3044

Interactive Question:
http://www.quizover.com/question/extending-java-abstract-class?pdf=3044

4.1.6. Which of the following is true?

Author: Yasser Ibrahim

Which of the following is true?

Please choose all the answers that apply:

- If a class is declared abstract , it must contain at least one abstract method
- If a class is declared abstract , all its methods must be abstract
- A method can either be final or abstract
- If a method is abstract then its class must be declared abstract
- An Abstract method has no body and ends with a semicolon

Check the answer of this question online at QuizOver.com:
Question: Rules of declaring java abstract Class

Flashcards:
http://www.quizover.com/flashcards/rules-of-declaring-java-abstract-class?pdf=3044

Interactive Question:
http://www.quizover.com/question/rules-of-declaring-java-abstract-class?pdf=3044

4.1.7. Which of the following variables is incorrectly declared?

Author: JavaChamp Team

Which of the following variables is incorrectly declared?

```
public abstract interface Bouncable {

 int a = 0;
 public int b = 1;
 public static int c = 2;
 public static transient int d = 3;
 public final int e = 3;
 public static final int f = 3;

}
```

Please choose only one answer:

- a
- b
- c
- d
- e
- f

Check the answer of this question online at QuizOver.com:
Question: Java Interface Variables

Flashcards:
http://www.quizover.com/flashcards/java-interface-variables?pdf=3044

Interactive Question:
http://www.quizover.com/question/java-interface-variables?pdf=3044

4.1.8. Three of the methods are incorrectly declared, which are they?

Author: JavaChamp Team

Three of the methods are incorrectly declared, which are they?

```
public abstract class Tester {
 public void test1();
 public final void test2() {};
 public static void test3() {};
 public abstract static void test4();
 public abstract final void test5();
}
```

Please choose only one answer:

- test1 , test2 and test4
- test2 , test4 and test5
- test1 , test4 and test5

Check the answer of this question online at QuizOver.com:
Question: How to declare java abstract method?

Flashcards:
http://www.quizover.com/flashcards/how-to-declare-java-abstract-method?pdf=3044

Interactive Question:
http://www.quizover.com/question/how-to-declare-java-abstract-method?pdf=3044

4.1.9. Considering the following declaration for interface Convertable, wh...

Author: Yasser Ibrahim

Considering the following declaration for interface Convertable, which of the following code segments will compile?

```
public interface Convertable {

 int convertToInt();
 char convertToChar();

}
```

Please choose all the answers that apply:

-
  ```
  class Digit implements Convertable {
  public char convertToChar() {
  return 0;
  }
  public int convertToInt() {
  return 0;
  }
  }
  ```

-
  ```
  abstract class Digit implements Convertable {
  int convertToInt() ;
  char convertToChar();
  }
  ```

-
  ```
  abstract class Digit implements Convertable {

   public int convertToInt() {
    return 0;
   }
  }
  ```

-
  ```
  abstract class Digit implements Convertable {
    public int convertToInt() {
     return 0;
    }
    char convertToChar();
  }
  ```

-
  ```
  class Digit implements Convertable {
    int convertToInt() {
     return 0;
    }

   char convertToChar() {
  ```

```
  return 0;
 }
}
```

-
  ```
  interface Roundable extends Convertable {
   int roundUp();
  }
  ```

Check the answer of this question online at QuizOver.com:
Question: What are the rules to implement an interface?

Flashcards:
http://www.quizover.com/flashcards/what-are-the-rules-to-implement-an-interface?pdf=3044

Interactive Question:
http://www.quizover.com/question/what-are-the-rules-to-implement-an-interface?pdf=3044

4.1.10. Which of the following declaration will compile without errors?

Author: JavaChamp Team

Which of the following declaration will compile without errors?

Please choose all the answers that apply:

- public abstract class Digit { public abstract void print(); }
- public class Digit { public abstract void print(); }
- public abstract class Digit { public abstract void print(){} }
- public abstract class Digit { public void print();}
- public class Digit { public void print(){};}

Check the answer of this question online at QuizOver.com:
Question: How to declare java abstract class?

Flashcards:
http://www.quizover.com/flashcards/how-to-declare-java-abstract-class?pdf=3044

Interactive Question:
http://www.quizover.com/question/how-to-declare-java-abstract-class?pdf=3044

4.1.11. Which of the following is correct about an interface in java?

Author: Yasser Ibrahim

Which of the following is correct about an interface in java?

Please choose all the answers that apply:

- An Interface can declare data members but must be constants
- All methods in an interface are implicitly abstract
- Methods in an interface may be public, private, protected.
- An interface can implement an abstract class
- An interface can be final

Check the answer of this question online at QuizOver.com:
Question: how to write an interface?

Flashcards:
http://www.quizover.com/flashcards/how-to-write-an-interface?pdf=3044

Interactive Question:
http://www.quizover.com/question/how-to-write-an-interface?pdf=3044

4.1.12. To resolve the compilation error(s) in the following code, what can...

Author: Yasser Ibrahim

To resolve the compilation error(s) in the following code, what can be done independently :

```
interface Movable {

 public abstract void m1(); // line 1

 void m2();  // line 2

 public void m3();  // line 3

 abstract void m4();  // line 4
}

class Chair implements Movable {  // line 5

 public void m1() { }  // line 6

 void m2() { }  // line 7
}
```

Please choose all the answers that apply:

- mark class Chair "abstract"
- mark Chair "abstract" and mark m2() "public"
- implement m3() and m4() in Chair (with public access modifier)
- implement the methods m3() and m4() in Chair (with public access modifier) and mark m2() in Chair "public"

Check the answer of this question online at QuizOver.com:
Question: how to implement an interface in java?

Flashcards:
http://www.quizover.com/flashcards/how-to-implement-an-interface-in-java?pdf=3044

Interactive Question:
http://www.quizover.com/question/how-to-implement-an-interface-in-java?pdf=3044

4.1.13. What is the correct output?

Author: MrDick

What is the correct output?

```
public interface Father {

    public void go();
}


 public interface Mother {

    public void go();
}

 class  Test implements Father, Mother   { // line 1

    @Override
 public void go(){   // line 2
            System.out.println("Child");
        }

 public static void main(String args[]) {

  new Test().go(); // line 3
 }
}
```

Please choose only one answer:

- Compilation error in line 1
- compliation error in line 2
- Child
- compilation error in line 3

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

Flashcards:
http://www.quizover.com/flashcards/java-interfaces?pdf=3044

Interactive Question:
http://www.quizover.com/question/java-interfaces?pdf=3044

4.1.14. Correct the following code:

Author: AlinaIoana Florea

Correct the following code:

```
interface Pet
{
   int MAX_PET_NR = 10;
   void play();
}

interface Dressable
{
   void dress();
}

interface Animal {} // line 1

abstract class Dog extends Animal, Dressable, Pet // line 2
{
   public void play(int min){}
   void dress(){} // line 3
   void bark(){}
}

class Bulldog extends Dog implements Pet
{
   Dog dog;
   Dog dog1 = new Dog(); // line 4
   Dog dog2 = new Bulldog();

   void bark() throws RuntimeException // line 5
   {
      MAX_PET_NR = 2; // line 6
   }; // line 5
}
```

Please choose all the answers that apply:

- line 1 - a class(abstract or concrete) can not be empty
- line 2 - interfaces must be "implemented" by non-interface classes, and "extended" only by other interfaces
- line 3 - to be a valid implementation/override, dress() must stay at least the same as available, so it must keep the "public" access with which was marked automatically in its interface
- line 4 - is not ok as Dog is abstract so might not contain fully implemented methods from which to create a whole object. Also Bulldog has to implement play() as the version inherited from Dog is an overload not an override.
- line 5 - to be a valid override barck() may not throw any exception if the class it overrides doesn't throw an exception. Nor is it valid for Bulldog to extend again Pet as it inherits it already from Dog.

Check the answer of this question online at QuizOver.com:
Question: how to implement java interfaces?

Flashcards:
http://www.quizover.com/flashcards/how-to-implement-java-interfaces?pdf=3044

Interactive Question:
http://www.quizover.com/question/how-to-implement-java-interfaces?pdf=3044