

# SCJP Mock Exams by JavaChamp.com

Open Certification Plattform

Authors: N. Ibrahim, Y. Ibrahim

Copyright (c) 2009

# Introducing JavaChamp.com Website

JavaChamp.com is a Java Open Certification Platform.

It provides an interactive web interface for the Java community to learn, interact and certify its acquired java experience.

JavaChamp.com helps the Java developers to achieve the highest impact of thier learning activities.

Start JavaChamp.com online exams now and prepare yourself for the next interview or certification!

Visit <http://www.JavaChamp.com>



- ↓ Services
- ↓ About us
- ↓ Contac Us

- Home
- Take Exam
- Log in
- Register
- Book

Search



→ Stop

## ★ About us

JavaChamp.com is a Java Open Certification Plattform!

- It provides unique services to help you develop yourself in the field of computer science and programming
- It provides a web interface for the Java community to learn,interact and certify its acquired java experience.
- JavaChamp.com helps the Java developers to achieve the

## ★ Why to register in our community?

**Registration is optional and free but brings many advantages**

- By registering in JavaChamp.com you can save and keep track of your exams for later revision and hence to help you monitor your progress.

## CHECK OUR eBook

JavaChamp.com's eBook encompasses a large number of multiple questions, which cover the subjects java,C and assembly. The book doesn't ...

More →

## ★ Exam in Session

Question 3 / 20

What is the expected output?

```
01. public class OuterTest {
02.
03.     public static void main(String args[]) {
04.         Airplane.BlackBox box = new Airplane().new BlackBox(); // line 1
05.         box.printVariables();
06.
07.     }
08. }
09.
10. class Airplane {
11.     String code = "11";
12.
13.     class BlackBox {
14.         String code = "22";
15.
16.         public void printVariables() {
17.             System.out.print(code);
18.             System.out.print(Airplane.this.code); // line 20
19.
20.         }
21.     }
22.
23. }
```

- Compile error because of line 1 (incorrect instantiation)
- Compile error because of line 20 (can't access Airplane's variables)
- 2222
- 1111
- 2211

[Back](#) [Next](#)[Finish and evaluate](#) [Abort Exam](#)

### CHECK OUR eBook

JavaChamp.com's eBook encompasses a large number of multiple questions, which cover the subjects java, C and assembly. The book doesn't ...

[More](#) →

# Copyright

Copyright 2009 JavaChamp.com

Online version published by JavaChamp.com Germany.

## DISCLAIMER

All services and content of JavaChamp.com are provided under JavaChamp.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 JavaChamp.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 JavaChamp.com is authorized hereunder except under this disclaimer.

The detailed "terms of use" of JavaChamp.com can be found under:

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

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 license.

The full license legal code can be found under:

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

And a human-readable summary of the this license can be found under:

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

According to the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 license You agree to the following:

You are free to share, copy, distribute and transmit the work under the following conditions:

- You must attribute the work to JavaChamp.com with a link to <http://www.javachamp.com>.
- You may not use this work for commercial purposes.
- You may not alter, transform, or build upon this work.

# Table of Contents

## 1. Java Programming Language

- Strings
- Constructors
- Inner class
- Flow Control
- Declarations and Access Control
- Interfaces and Abstract Classes
- Exceptions
- Inheritance
- Arrays
- Operators
- Variables and Methods
- Wrappers and Autoboxing
- Garbage Collection
- Overriding and Overloading
- Collections
- Generics
- Formatting
- I/O
- Threads
- Enums
- Data Types
- Static and init blocks
- Serialization

# 1. Chapter: Java Programming Language

## Chapter Description and Objectives

### 1. Generics

#### Exam Category Description and Objectives

##### 1.1.1. How to declare instantiate java generic collection?

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

```
import java.util.ArrayList;
import java.util.List;

public class Tester {

    public static void main(String[] args) {

        List<String> list1 = new ArrayList<String>();//line 1
        List<Object> list2 = list1;//line 2
        list2.add(new Integer(12));//line 3
        System.out.println(list2.size());//line 4

    }
}
```

Please choose only one answer:

- Compilation error at line 1, must initialize the size
- Compilation error at line 2, Type mismatch, cannot convert from List<String> to List<Object>
- Compilation error at line 3, can add objects of type String only
- 1

Check this Question online on JavaChamp.com: [How to declare instantiate java generic collection?](#)

### 1.1.2. How to add to java generic collection?

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

```
import java.util.ArrayList;
import java.util.List;

interface chewable {}

class Gum implements chewable {}

public class Tester {

    public static void main(String[] args) {
        List<Gum> list1 = new ArrayList<Gum>();
        list1.add(new Gum());
        List list2 = list1;
        list2.add(new Integer(9));
        System.out.println(list2.size());
    }
}
```

Please choose only one answer:

- Compilation error
- 2 will be printed but with warnings
- 2 will be printed without warnings
- An exception will be thrown at runtime

Check this Question online on JavaChamp.com: [How to add to java generic collection?](#)

### 1.1.3. How to declare java generic collection?

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

```
import java.util.ArrayList;
import java.util.List;

interface chewable {}

class Gum implements chewable {}

class Meat implements chewable {}

public class Tester {

    public static void main(String[] args) {
        List list1 = new ArrayList<chewable>(); // line 1
        list1.add(new Gum()); // line 2
        list1.add(new Meat()); // line 3
        list1.add(new Integer(9)); // line 4
        System.out.println(list1.size()); // line 5
    }
}
```

Please choose only one answer:

- Compilation error at line 1
- Compilation error at line 4
- Runtime exception thrown at line 1
- Runtime exception thrown at line 4
- 3

Check this Question online on JavaChamp.com: [How to declare java generic collection?](#)

#### 1.1.4. java generic collection rules?

Which of the following is a correct declaration and instantiation of list?

Please choose all the answers that apply:

- `List<String> list = new ArrayList<String>();`
- `List<?> list = new ArrayList<String>();`
- `List<? extends Object> list = new ArrayList<String>();`
- `List<Object> list = new ArrayList<String>();`
- `List list = new ArrayList<String>();`
- `List list = new ArrayList<?>();`

Check this Question online on JavaChamp.com: [java generic collection rules?](#)

### 1.1.5. When cannot add to a generic collection?

The following code contains a compilation error, at what line?

```
import java.util.ArrayList;
import java.util.List;

interface Chewable {}
class Meat implements Chewable {}

public class Tester {

    public static void main(String[] args) {

        List<? extends Chewable> list1 = new ArrayList<Meat>(); // Line 11

        List<Chewable> list2      = new ArrayList<Chewable>(); // Line 13

        Meat meat = new Meat();

        list1.add(meat); // Line 17

        list2.add(meat); // Line 19
    }
}
```

Please choose only one answer:

- Line 11
- Line 13
- Line 17
- Line 19

Check this Question online on JavaChamp.com: [When cannot add to a generic collection?](#)

### 1.1.6. Proper instantiation of java generic collection

The following program contains two compilation errors, at what lines?

```
import java.util.ArrayList;
import java.util.List;

interface Chewable {}
class Meat implements Chewable {}

public class Tester {

    public static List<? extends Chewable> printSize(List<? extends Chewable> list) { // Line 10
        System.out.println(list.size());
        return list;
    }
    public static void main(String[] args) {

        List<? extends Chewable> list1 = new ArrayList<Meat>(); // Line 16
        List<Chewable> list2 = new ArrayList<Chewable>(); // Line 17
        List<Meat> list3 = new ArrayList<Meat>(); // Line 18

        list1 = printSize(list1); // Line 20
        list2 = printSize(list2); // Line 21
        list3 = printSize(list3); // Line 22

    }
}
```

Please choose all the answers that apply:

- Line 10
- Line 16
- Line 17
- Line 18
- Line 20
- Line 21
- Line 22

Check this Question online on JavaChamp.com: [Proper instantiation of java generic collection](#)

### 1.1.7. When were Generics first introduced in Java?

When were generics first introduced in Java?

Please choose only one answer:

- in JDK 1.4
- in JDK 1.5
- in JDK 1.6

Check this Question online on JavaChamp.com: [When were Generics first introduced in Java?](#)

### 1.1.8. how to sort a generic collection?

What is the expected output of compiling and running the following code?

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Test{
    public static void main(String[] args) {

        List<Foo> myFooList = new ArrayList<Foo>();
        myFooList.add(new Foo("C"));
        myFooList.add(new Foo("A"));
        myFooList.add(new Foo("D"));

        Collections.sort(myFooList);
        System.out.print(myFooList.get(0).code);

    }
}

class Foo implements Comparable<Foo> {
    String code;

    Foo(String c) {
        code = c;
    }

    int compareTo(Foo f) {
        return this.code.compareTo(f.code);
    }
}
```

Please choose only one answer:

- A
- C
- D
- no output is printed
- compilation error
- exception thrown at run time

Check this Question online on JavaChamp.com: [how to sort a generic collection?](https://www.java-champ.com/question/collections/how-to-sort-a-generic-collection/)

### 1.1.9. inserting into a non-generic collection in java

What is the expected output of compiling and running the following code?

```
import java.util.ArrayList;
import java.util.List;

public class Test{
    public static void main(String[] args) {

        List list = new ArrayList();

        list.add("Hello");
        Foo f = new Foo();
        list.add(f);

        f = list.get(1);
        System.out.print(list.get(0) + "-" + f);

    }
}

class Foo {
    public String toString() {
        return "Foo";
    }
}
```

Please choose only one answer:

- Hello-Foo
- compilation error
- exception at run time

Check this Question online on JavaChamp.com: [inserting into a non-generic collection in java](https://www.java-champ.com/question/inserting-into-a-non-generic-collection-in-java/)

### 1.1.10. generic collection as return type in java

Considering that all needed imports are there, what, inserted independently at line 1, will compile? (choose all correct answers)

```
class A {}
class B extends A{}
class C extends B{}

public class Test {

    public List<? extends B> ring(){
// line 1
    }
}
```

Please choose all the answers that apply:

- return new ArrayList<B>();
- return new ArrayList<C>();
- return new ArrayList<A>();
- return new ArrayList<? extends B>();
- return new LinkedList<B>();
- return new PriorityQueue<B>();
- return new ArrayList();

Check this Question online on JavaChamp.com: [generic collection as return type in java](#)

### 1.1.11. generic methods in java

Will the following code compile?

```
import java.util.*;

class B {
}

class C extends B {
}

public class Test {

    public static <E extends B> Queue<E> ring(List<E> list) {
        return null;
    }

    public static void main(String[] args) {
        List<C> list1 = new ArrayList<C>();
        ArrayList<C> list2 = new ArrayList<C>();

        Queue<C> q1;
        PriorityQueue<C> q2;

        q1 = ring(list1); // line1
        q1 = ring(list2); // line2
        q2 = ring(list1); // line3
        q2 = ring(list2); // line4
    }
}
```

Please choose all the answers that apply:

- Yes, the program is free of compilation errors
- No, there is a compilation error at // Line 1
- No, there is a compilation error at // Line 2
- No, there is a compilation error at // Line 3
- No, there is a compilation error at // Line 4

Check this Question online on JavaChamp.com: [generic methods in java](#)

### 1.1.12. how to declare a generic collection in java?

How can this program be modified to make use of appropriate generic types? choose 3 answers (one modification for each line)

```
import java.util.*;

public class Test {

    public static void main(String[] args) {
        List ids = new ArrayList(); // Line 1
        ids.add(123);
        ids.add(999);

        Map students = new HashMap(); // Line 2
        students.put("Jess",ids.get(0));
        students.put("Jimmy",ids.get(1));

        int x = ((Long)students.get("Jimmy")).intValue(); // Line 3
    }
}
```

Please choose all the answers that apply:

- replace line 1 with `List<Integer> ids = new ArrayList<Integer>();`
- replace line 1 with `List<Long> ids = new ArrayList<Long>();`
- replace line 2 with `Map<Integer,String> students = new HashMap<Integer,String>();`
- replace line 2 with `Map<String,Integer> students = new HashMap<String,Integer>();`
- replace line 3 with `int x = students.get("Jimmy");`
- replace line 3 with `Long x = students.get("Jimmy");`

Check this Question online on JavaChamp.com: [how to declare a generic collection in java?](https://www.javachamp.com/question/how-to-declare-a-generic-collection-in-java/)

### 1.1.13. how to declare a generic Map in java?

Given the following incomplete program, based on the given put statements in map, what is considered a correct declaration and instantiation for map ?

```
import java.util.*;

public class Test {

    public static void main(String[] args) {

        // insert code here

        map.put(new ArrayList<Integer>(), 1);
        map.put(new ArrayList<Integer>(), 12);
        map.put(new LinkedList<Integer>(), new Integer(1));
        map.put(new LinkedList<Integer>(), new Long(1));

    }
}
```

Please choose only one answer:

- Map<List<Integer>,? extends Number> map = new HashMap<List<Integer>,Number>();
- Map<List<Integer>,Number> map = new HashMap<List<Integer>,Integer>();
- Map<List<Integer>, Number> map = new HashMap<List<Integer>, Number>();
- Map<List<Integer>,Integer> map = new HashMap<List<Integer>,Integer>();
- Map<ArrayList<Integer>,Integer> map = new HashMap<ArrayList<Integer>,Integer>();
- Map<List<Integer>,Long> map = new HashMap<List<Integer>,Long>();

Check this Question online on JavaChamp.com: [how to declare a generic Map in java?](https://www.java-champ.com/question/generics/how-to-declare-a-generic-map-in-java/)

