

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

SCJP Dumps Doc 10 of 16

This document is Tenth in series of 16 documents for Java SCJP Dumps. Links to all documents are available in Facebook group [JAVA Tutorials](#). Join the group and download the documents.

Index : Download Links for other Doc :

JAVA OCPJP Dumps Part 1 : [Language Fundamentals](#)

JAVA OCPJPDumps Part 2..1 : [Object Oriented Programming.](#)

JAVA OCPJPDumps Part 2.2 : [Object Oriented Programming.](#)

JAVA OCPJPDumps Part 3 : [Declarations & Access Control](#)

JAVA OCPJPDumps Part 4 : [Operators](#)

JAVA OCPJPDumps Part 5 : [Flow Control](#)

JAVA OCPJPDumps Part 6 : [Assertions](#)

JAVA OCPJPDumps Part 7 : [Exception Handling](#)

JAVA OCPJPDumps Part 8 : [Garbage Collections](#)

JAVA OCPJPDumps Part 9 : [Multi-Threading](#)

JAVA OCPJPDumps Part 10 : [Collections & Generics](#)

JAVA OCPJPDumps Part 11 : [JAVA Packages](#)

JAVA OCPJPDumps Part 12 : [Development](#)

JAVA OCPJPDumps Part 13 : [Internationalization](#)

JAVA OCPJPDumps Part 14 : [JAVA I/O & Serialization](#)

JAVA OCPJPDumps Part 15 : [Inner Classes](#)

JAVA OCPJPDumps Part 16 : [Miscellaneous Questions](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part 1](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part2](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part3](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part4](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part5](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part6](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part7](#)

Collections and Generics

Q: 01 Given:

```
34. HashMap props = new HashMap();
35. props.put("key45", "some value");
36. props.put("key12", "some other value");
37. props.put("key39", "yet another value");
38. Set s = props.keySet();
39. // insert code here
```

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

What, inserted at line 39, will sort the keys in the props HashMap?

- A. Arrays.sort(s);
- B. s = new TreeSet(s);
- C. Collections.sort(s);
- D. s = new SortedSet(s);

Answer: B

Q: 02 Click the Exhibit button.

```
1. import java.util.*;
2. public class TestSet {
3.     enum Example { ONE, TWO, THREE }
4.     public static void main(String[] args)
5.     {
6.         Collection coll = new ArrayList();
7.         coll.add(Example.THREE);
8.         coll.add(Example.THREE);
9.         coll.add(Example.THREE);
10.        coll.add(Example.TWO);
11.        coll.add(Example.TWO);
12.        coll.add(Example.ONE);
13.        Set set = new HashSet(coll);
14.    }
```

Which statement is true about the set variable on line 12?

- A. The set variable contains all six elements from the coll collection, and the order is guaranteed to be preserved.
- B. The set variable contains only three elements from the coll collection, and the order is guaranteed to be preserved.
- C. The set variable contains all six elements from the coll collection, but the order is NOT guaranteed to be preserved.
- D. The set variable contains only three elements from the coll collection, but the order is NOT guaranteed to be preserved.

Answer: D

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 03 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. Compilation fails due to an error in line 23.
- B. Compilation fails due to an error in line 29.
- C. A ClassCastException occurs in line 29.
- D. A ClassCastException occurs in line 31.
- E. The value of all four objects prints in natural order.

Answer: C

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 04 Click the Task button.

Place code into the class so that it compiles and generates the output
answer=42 . Note: Code options may be used more than once.

Class

```
public class Place here {  
    private Place here object;  
    public Place here (Place here object) {  
        this.object = object;  
    }  
    public Place here getObject() {  
        return object;  
    }  
  
    public static void main(String[] args) {  
        Gen<String> str = new Gen<String>("answer");  
        Gen<Integer> intg = new Gen<Integer>(42);  
        System.out.println(str.getObject() + "=" +  
            intg.getObject());  
    }  
}
```

Code Options

Gen<T>

Gen<?>

Gen

?

T

Done

Solution:

- 1.Gen<T>
- 2.T
- 3.Gen
- 4.T
- 5.T

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 05 Click the Task button.

Given:

```
public void takeList(List<? extends String> list) {  
    // insert code here  
}
```

Place the Compilation Results on each code statement to indicate whether or not that code will compile if inserted into the takeList() method.

Code Statements

`list.add("Foo");`

`list = new ArrayList<String>();`

`list = new ArrayList<Object>();`

`String s = list.get(0);`

`Object o = list;`

Compilation Results

Compilation succeeds

Compilation fails

Done

Solution:

1. `list.add("foo");` ----- Compilation fails
2. `list = new ArrayList<String>();` -----Compilation succeeds
3. `list=new ArrayList<Object>();` ---- Compilation fails
4. `String s = list.get(0);` ----- Compilation succeeds
5. `Object o = list;` ----- Compilation succeeds

Q: 06 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. `}`

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Which statement is true?

- A. The equals method does NOT properly override the Object.equals method.
- B. Compilation fails because the private attribute p.name cannot be accessed in line 5.
- C. To work correctly with hash-based data structures, this class must also implement the hashCode method.
- D. When adding Person objects to a java.util.Set collection, the equals method in line 4 will prevent duplicates.

Answer: A

Q: 07 Given:

```
1. import java.util.*;  
2. public class Old {  
3.     public static Object get0(List list) {  
4.         return list.get(0);  
5.     }  
6. }
```

Which three will compile successfully? (Choose three.)

- A. Object o = Old.get0(new LinkedList());
- B. Object o = Old.get0(new LinkedList<?>());
- C. String s = Old.get0(new LinkedList<String>());
- D. Object o = Old.get0(new LinkedList<Object>());
- E. String s = (String)Old.get0(new LinkedList<String>());

Answer: A, D, E

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 08 Given:

```
1. import java.util.*;
2. public class Example {
3. public static void main(String[] args) {
4. // insert code here
5. set.add(new Integer(2));
6. set.add(new Integer(1));
7. System.out.println(set);
8. }
9. }
```

Which code, inserted at line 4, guarantees that this program will output [1, 2]?

- A. Set set = new TreeSet();
- B. Set set = new HashSet();
- C. Set set = new SortedSet();
- D. List set = new SortedList();
- E. Set set = new LinkedHashSet();

Answer: A

Q: 09 Given:

```
11. public static Collection get() {
12. Collection sorted = new LinkedList();
13. sorted.add("B"); sorted.add("C"); sorted.add("A");
14. return sorted;
15. }
16. public static void main(String[] args) {
17. for (Object obj: get()) {
18. System.out.print(obj + ", ");
19. }
20. }
```

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

What is the result?

- A. A, B, C,
- B. B, C, A,
- C. Compilation fails.
- D. The code runs with no output.
- E. An exception is thrown at runtime.

Answer: B

Q:10 given

Click the Task button.

Place the correct description of the compiler output on the code fragments to be inserted at lines 4 and 5. The same compiler output may be used more than once.

```
1. import java.util.*;
2. public class X {
3.     public static void main(String[] args) {
4.         // insert code here
5.         // insert code here
6.     }
7.     public static void foo(List<Object> list) {
8.     } }
```

Code

ArrayList<String> x1 = new ArrayList<String>();
foo(x1);

ArrayList<Object> x2 = new ArrayList<String>();
foo(x2);

ArrayList<Object> x3 = new ArrayList<Object>();
foo(x3);

ArrayList x4 = new ArrayList();
foo(x4);

Compiler Output

Compilation succeeds.

Compilation fails due to an error in the first statement.

Compilation of the first statement succeeds, but compilation fails due to an error in the second statement.

Done

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Solution:

Compilation of the first statement succeeds ,but compilation fails due to an error in the second statement.

1. **Compilation fails due to an error in the first statement**
2. **Compilation succeeds**
4. **Compilation succeeds**

Q: 11 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. 3, 2, 1,
- B. B. 1, 2, 3,
- C. Compilation fails.
- D. The code runs with no output.
- E. An exception is thrown at runtime.

Answer: C

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 12 Click the Task button

Given:

```
1. import java.util.*;
2. class A { }
3. class B extends A { }
4. public class Test {
5.     public static void main(String[] args) {
6.         List<A> listA = new LinkedList<A>();
7.         List<B> listB = new LinkedList<B>();
8.         List<Object> listO = new LinkedList<Object>();
9.         // insert code here
10.    }
11.    public static void m1(List<? extends A> list) { }
12.    public static void m2(List<A> list) { }
13. }
```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

Method Calls		Result
m1(listA);	m2(listA);	Does not compile.
m1(listB);	m2(listB);	Compiles and runs without error.
m1(listO);	m2(listO);	An exception is thrown at runtime.

=====does not compile=====

1.m1(listO);

2.m2(listB);

3.m2(listO);

=====compiles and runs with out error=====

1.m1(listA);

2.m1(listB);

3.m2(listA);

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 13 Given:

```
1. import java.util.*;
2. public class PQ {
3. public static void main(String[] args) {
4. PriorityQueue<String> pq = new PriorityQueue<String>();
5. pq.add("carrot");
6. pq.add("apple");
7. pq.add("banana");
8. System.out.println(pq.poll() + ":" + pq.peek());
9. }
10. }
```

What is the result?

- A. apple:apple
- B. carrot:apple
- C. apple:banana
- D. banana:apple
- E. carrot:carrot
- F. carrot:banana

Answer: C

Q: 14 Given:

```
1. import java.util.*;
2. public class WrappedString {
3. private String s;
4. public WrappedString(String s) { this.s = s; }
5. public static void main(String[] args) {
6. HashSet<Object> hs = new HashSet<Object>();
7. WrappedString ws1 = new WrappedString("aardvark");
8. WrappedString ws2 = new WrappedString("aardvark");
9. String s1 = new String("aardvark");
10. String s2 = new String("aardvark");
11. hs.add(ws1); hs.add(ws2); hs.add(s1); hs.add(s2);
12. System.out.println(hs.size()); } }
```

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

What is the result?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. Compilation fails.
- G. An exception is thrown at runtime.

Answer: D

Q: 15 Given:

```
11. public class Key {  
12.     private long id1;  
13.     private long id2;  
14.  
15. // class Key methods  
16. }
```

A programmer is developing a class Key, that will be used as a key in a standard java.util.HashMap. Which two methods should be overridden to assure that Key works correctly as a key? (Choose two.)

- A. public int hashCode()
- B. public boolean equals(Key k)
- C. public int compareTo(Object o)
- D. public boolean equals(Object o)
- E. public boolean compareTo(Key k)

Answer: A, D

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 16 Given a pre-generics implementation of a method:

```
11. public static int sum(List list) {  
12.     int sum = 0;  
13.     for ( Iterator iter = list.iterator(); iter.hasNext(); ) {  
14.         int i = ((Integer)iter.next()).intValue();  
15.         sum += i;  
16.     }  
17.     return sum;  
18. }
```

Which three changes must be made to the method sum to use generics?

(Choose three.)

- A. remove line 14
- B. replace line 14 with "int i = iter.next();"
- C. replace line 13 with "for (int i : intList) {"
- D. replace line 13 with "for (Iterator iter : intList) {"
- E. replace the method declaration with "sum(List<int> intList)"
- F. replace the method declaration with "sum(List<Integer> intList)"

Answer: A, C, F

Q: 17 Given:

```
11. // insert code here  
12. private N min, max;  
13. public N getMin() { return min; }  
14. public N getMax() { return max; }  
15. public void add(N added) {  
16.     if (min == null || added.doubleValue() < min.doubleValue()) 17. min = added;  
18.     if (max == null || added.doubleValue() > max.doubleValue()) 19. max =  
added;  
20. }  
21. }
```

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Which two, inserted at line 11, will allow the code to compile? (Choose two.)

- A. public class MinMax<?> {
- B. public class MinMax<? extends Number> {
- C. public class MinMax<N extends Object> {
- D. public class MinMax<N extends Number> {
- E. public class MinMax<? extends Object> {
- F. public class MinMax<N extends Integer> {

Answer: D, F

Q: 18 Given:

```
1. import java.util.*;
2.
3. public class LetterASort{
4.     public static void main(String[] args) {
5.         ArrayList<String> strings = new ArrayList<String>();
6.         strings.add("aAaA");
7.         strings.add("AaA");
8.         strings.add("aAa");
9.         strings.add("AAaa");
10.        Collections.sort(strings);
11.        for (String s : strings) { System.out.print(s + " "); }
12.    }
13. }
```

What is the result?

- A. Compilation fails.
- B. aAaA aAa AAaa AaA
- C. AAaa AaA aAa aAaA
- D. AaA AAaa aAaA aAa
- E. aAa AaA aAaA AAaa
- F. An exception is thrown at runtime.

Answer: C

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 19 Click the Task button.

Given: `NumberNames nn = new NumberNames();`
`nn.put("one", 1);`
`System.out.println(nn.getNames());`



Place the code into position to create a class that maps from Strings to integer values. The result of execution must be [one]. Some options may be used more than once.

```
public class NumberNames {  
    private HashMap<Place here , Place here > map =  
        new HashMap<Place here , Place here Place here > ;  
    public void put(String name, int value) {  
        map.put(Place here , Place here );  
    }  
    public Place here getNames() {  
        return map.keySet();  
    }  
}
```

Code

Set<int>	Set<Integer>	HashSet
Set<Integer,String>	Set<int, String>	Set<String,Integer>
Set<String, int>	Set<String>	NumberNames
String	Integer	int
>()	name	value
		map

Done

Solution:

```
public class NumberNames{  
    private HashMap< String , Integer> map=  
    new HashMap< String , Integer >( );  
    public void put(String name , int Value) {  
        map.put(name , Value);  
    }  
    public Set<String> getNames( ) {  
        return map.keySet( );  
    }  
}
```

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 20 Which two statements are true about the hashCode method? (Choose two.)

- A. The hashCode method for a given class can be used to test for object equality and object inequality for that class.
- B. The hashCode method is used by the java.util.SortedSet collection class to order the elements within that set.
- C. The hashCode method for a given class can be used to test for object inequality, but NOT object equality, for that class.
- D. The only important characteristic of the values returned by a hashCode method is that the distribution of values must follow a Gaussian distribution.
- E. The hashCode method is used by the java.util.HashSet collection class to group the elements within that set into hash buckets for swift retrieval.

Answer: C, E

Q: 22 A programmer has an algorithm that requires a java.util.List that provides an efficient implementation of add(0, object), but does NOT need to support quick random access.

What supports these requirements?

- A. java.util.Queue
- B. java.util.ArrayList
- C. java.util.LinearList
- D. java.util.LinkedList

Answer: D

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 24 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. The code runs with no output.
- B. An exception is thrown at runtime.
- C. Compilation fails because of an error in line 20.
- D. Compilation fails because of an error in line 21.
- E. Compilation fails because of an error in line 23.
- F. Compilation fails because of an error in line 25.

Answer: F

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Q: 26 Given:

Place the correct description of the compiler output on the code fragments to be at lines 4 and 5. The same compiler output may be used more than once.

```
1. import java.util.*;
2. public class X {
3.     public static void main(String[] args) {
4.         // insert code here
5.         // insert code here
6.     }
7.     public static void foo(List<Object> list) {
8.     } }
```



Code

```
ArrayList<String> x1 = new ArrayList<String>();
foo(x1);
```

```
ArrayList<Object> x2 = new ArrayList<String>();
foo(x2);
```

```
ArrayList<Object> x3 = new ArrayList<Object>();
foo(x3);
```

```
ArrayList x4 = new ArrayList();
foo(x4);
```

Compiler Output

Compilation succeeds.

Compilation fails due to an error in the first statement.

Compilation of the first statement succeeds, but compilation fails due to an error in the second statement.

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Solution:

1. **Compilation of the first statement succeeds ,but compilation fails due to an error in the second statement.**
2. **Compilation fails due to an error in the first statement**
3. **Compilation succeeds**
4. **Compilation succeeds**

Q: 26 Click the Task button.

Given:

```
public void takeList(List<? extends String> list) {  
    // insert code here  
}
```

Place the Compilation Results on each code statement to indicate whether or not that code will compile if inserted into the takeList() method.

Code Statements

`list.add("Foo");`

`list = new ArrayList<String>();`

`list = new ArrayList<Object>();`

`String s = list.get(0);`

`Object o = list;`

Compilation Results

Compilation succeeds

Compilation fails

Done

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Solution:

1. list.add("foo"); ----- Compilation fails
2. list = new ArrayList<String>(); -----Compilation succeeds
3. list=new ArrayList<Object>(); ---- Compilation fails
4. String s = list.get(0); ----- Compilation succeeds
5. Object o = list; ----- Compilation succeeds

Q: 27 Given:

1. public class Drink implements Comparable {
 2. public String name;
 3. public int compareTo(Object o) {
 4. return 0;
 5. }
 6. }
- and:
20. Drink one = new Drink();
 21. Drink two = new Drink();
 22. one.name= "Coffee";
 23. two.name= "Tea";
 23. TreeSet set = new TreeSet();
 24. set.add(one);
 25. set.add(two);

A programmer iterates over the TreeSet and prints the name of each Drink object.

What is the result?

- A. Tea
- B. Coffee
- C. Coffee Tea
- D. Compilation fails.
- E. The code runs with no output.
- F. An exception is thrown at runtime.

Answer: B

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Join [JAVA Tutorials](#) facebook group for more dumps : [Google +](#) group

Index : Download Links for other Doc :

JAVA OCPJP Dumps Part 1 : [Language Fundamentals](#)

JAVA OCPJPDumps Part 2..1 : [Object Oriented Programming.](#)

JAVA OCPJPDumps Part 2.2 : [Object Oriented Programming.](#)

JAVA OCPJPDumps Part 3 : [Declarations & Access Control](#)

JAVA OCPJPDumps Part 4 : [Operators](#)

JAVA OCPJPDumps Part 5 : [Flow Control](#)

JAVA OCPJPDumps Part 6 : [Assertions](#)

JAVA OCPJPDumps Part 7 : [Exception Handling](#)

JAVA OCPJPDumps Part 8 : [Garbage Collections](#)

JAVA OCPJPDumps Part 9 : [Multi-Threading](#)

JAVA OCPJPDumps Part 10 : [Collections & Generics](#)

JAVA OCPJPDumps Part 11 : [JAVA Packages](#)

JAVA OCPJPDumps Part 12 : [Development](#)

JAVA OCPJPDumps Part 13 : [Internationalization](#)

JAVA OCPJPDumps Part 14 : [JAVA I/O & Serialization](#)

JAVA OCPJPDumps Part 15 : [Inner Classes](#)

JAVA OCPJPDumps Part 16 : [Miscellaneous Questions](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part 1](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part2](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part3](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part4](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part5](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part6](#)

JAVA OCPJPMisc Dumps : [Misc Dump Part7](#)

JAVA Tutorials