



2. Generic methods are independent of the class's genericity.

3. What happens when we call it?

Notice there are two to three — one in the class, one in the method. They look the same, but they are completely independent.

```

public static void main(String[] args) {
    List<String> list = new ArrayList<String>();
    list.add("Hello");
    list.add("World");
}

public class Test {
    public void test() {
        String s = list.get(0);
    }
}

```

Left analysis step by step:

1. `list` —> the class is `List`, the object is `list`.
2. `list.get(0)` —> the method is `get`.

This just happens to be the same after, but they represent different type parameters.

So in this call:

- The class `T` = `String`
- The method `T` = `String`

So, the method returns a `String`, not a `Object`.

## Bounding Generic Types

Table 3.3 Type bounds

Type of bound	Syntax	Example
Unbound	<code>? extends T</code>	<code>List&lt;? extends Object&gt; list;</code>
Wildcard	<code>? extends T</code>	<code>List&lt;? extends String&gt; list = new ArrayList&lt;String&gt;();</code>
Wildcard with upper bound type	<code>? extends T</code>	<code>List&lt;? extends Exception&gt; list = new ArrayList&lt;Exception&gt;();</code>
Wildcard with lower bound type	<code>? super T</code>	<code>List&lt;? super Exception&gt; list = new ArrayList&lt;Object&gt;();</code>

What's wrong? A `String` is a subclass of an `Object`. This is true. However, `List<String>` cannot be assigned to `List<Object>`. We know it doesn't want `String` Java is trying to prevent us from ourselves with this one. Imagine if we could write code like this:

```

List<Integer> numbers = new ArrayList<Integer>();
numbers.add(Integer.valueOf(42));
List<Object> objects = numbers; // DOES NOT COMPILE
objects.add("Hello");
System.out.println(objects);

```

But the compiler prevents us that only `String` objects will appear in `numbers`. If this compiled, `list` would break that promise by putting a `String` there since `String` and `Object` are references to the same object. Good thing the compiler prevents this.

1. Big picture — what `extends` and `super` mean

2. The intuition behind `extends` (covariance)

3. The intuition behind `super` (contravariance)

4. Mnemonic: PECS

Producer — Extends  
Consumer — Super

- If your list produces data (you only read it), use `extends`.
- If your list consumes data (you only write into it), use `super`.

5. The "safety contract"

Here's what the compiler promises:

Wildcard	You can read	You can write	Example use
<code>? extends T</code>	As <code>T</code>	Not allowed	Reading elements (producer)
<code>? super T</code>	As <code>T</code>	As <code>T</code>	Writing elements (consumer)
<code>?</code>	As <code>Object</code>	Not allowed	Totaly generic (read+write)

9. `List<? extends B> listA = new ArrayList<A>;` // DOES NOT COMPILE  
10. `List<? super B> listB = new ArrayList<A>;`  
11. `List<?> listC = new ArrayList<Object>;` // DOES NOT COMPILE

Finally, `list` has no reference to any parent-type stored in it is an unbounded wildcard. The problem is that you need to know what type will be returned. `returning the wrapped` It wouldn't be useful anyway, because you can't add any elements to that `ArrayList`.

The confusing example

```

public void m() {
    String s = new Stack();
    return s;
}

// At first glance, it looks like we're just saying "create a new Object."
// Actually, any object supporting the interface of Object will do.

```

Step 2: Inside the method, `s` is now shadowing the real class `Object`. Suppose you already had this class hierarchy:

```

class O
class X extends O
class Y extends X

```

Now, when you write:

```

Object o = new Stack();

```

What happened? `o` is now `Stack`! Because the compiler doesn't know what specific the actual type is. If it could be a `Stack` or a `ArrayList` — on it plays safe and lets you only read it as `Object`.

Step 3: Why `m()` doesn't compile

How you might think: "If `m()` returns `s`, shouldn't I be able to use `s`?"

No — because at compile time, the compiler doesn't know which actual object of `A` it refers to.

- If you write `s`
- If you write `Object`
- If you write `String`

Since it's not a generic type variable, the compiler can't assume it has no argument constructor, or even that it's instantiated as `A`.

So the best:

```

Object o = new Stack();

```

So for the compiler:

- It's a `Stack` object
- It's a `String` object
- It's a `Object` object

So the compiler doesn't support the inclusion of the `Object` part.

Instead we could read:

Possible fix:

Step 4: The only thing you can safely do with `s` is use it as a type, not create it

For example:

```

Object o = new Stack();
return o; // This works fine.

```

Because `Stack` which contains objects of type `s`, so returning one is type-safe.

But creating one with `new(s)` is impossible, since generics are erased at runtime (the JVM doesn't know what `s` really is).

Difference between `B extends A` and `Extends A`.

Where they differ — scope and control

Aspect	<code>B extends A</code> (wildcard)	<code>Extends A</code> (generic parameter)
Where used	Only in type arguments (e.g., <code>List&lt;extends A&gt;</code> )	Decimated for a method or class (e.g., <code>Object extends A</code> )
Has a name?	No — anonymous placeholder	Yes — you can name <code>s</code> you can alias
Can refer to it elsewhere?	No — can't use <code>s</code> as a type, only <code>A</code> .	Yes, you can declare references, alias & etc.
Resolvable vs. wholtile	Resolvable (the solution — producer)	Fully resolved as a named type variable
Typical use	The method parameters of <code>Object</code> accept any type, so you can't constrain the type again.	For generic methods or classes that implement or extend the type.

Finally, can you figure out why this example does not compile?

```

Object o = new Stack();
List<Object> list = new ArrayList<Object>();
list.add(o); // DOES NOT COMPILE

```

This last method, `list.add()`, does not compile because it tries to mix a method-type parameter with a wildcard. A wildcard must have a `?` in it.

class A {  
 A() {}  
}  
A a = new A();  
List<A> list = new ArrayList<A>();  
list.add(a); // DOES NOT COMPILE

cific type parameter with a wildcard. A wildcard must have a ? in it.

class. A compiler error results from code that attempts to add an item to a list with an unbounded or upper bounded wildcard.

## EXAM

1. A, F !! When says "Additionally" means 2 scenarios and 2 ans.

2. G, C

3. Which of the following are true? (Choose all that apply)

```
12: List<String> s = list.of("mouse", "java");
13: var x = s.list.of("mouse", "java");
14:
15: s.removal(0);
16: s.removalIf(x.length() >= 4);
17: s.removeIf(x -> x.length() == 4);
18: s.removeIf(x -> x.length() == 4);
```

A. This code compiles and runs without error.  
B. Exactly one of these lines contains a compiler error.

C. Exactly three of these lines contain a compiler error.

D. Exactly four of these lines contain a compiler error.

E. If any lines with compiler errors are removed, this code runs without an runtime exception.

F. If all lines with compiler errors are removed, this code throws an exception.

3. b

3. What is the result of the following statement?

```
3: var greetings = new Arraylist<String>();
4: greetings.addFirst("Hello");
5: greetings.addLast("Hi");
6: greetings.addFirst("Hello");
7: greetings.addLast("Hi");
8: greetings.pop();
9: while (greetings.pop() != null)
10:     System.out.println(greetings.pop());
```

A. Hello

B. HelloHi

C. HelloHelloHi

D. Hi

E. The code does not compile.

F. An exception is thrown.

4. B,F

4. Which of these statements compile? (Choose all that apply)

- A. HashSet<Character> hs = new HashSet<Character>();
- B. HashSet<? super Comparable> set = new HashSet<Exception>();
- C. List<> list = new ArrayList<String>();
- D. List<Objects> values = new HashSet<Object>();
- E. List<Objects> objects = new ArrayList<? extends Object>();
- F. Map<String, ? extends Number> hm = new HashMap<String, Integer>();

5. E => B

5. What is the result of the following code?

```
5: public record HelloId(int id) {
6:     public HelloId() { this.id = 1; }
7:     public void print(String message) {
8:         System.out.println(id + " " + message);
9:     }
10: }
11: public static void main(String[] args) {
12:     new HelloId().print("Hello");
13:     new HelloId().print("World");
14: }
```

A. hi followed by a runtime exception.

B. hi-hello-world

C. The first compiler error is on line 1.

D. The second compiler error is on line 5.

E. The first compiler error is on the 6.

F. The first compiler error is on another line.

6. B,F

6. Which of the following use `Set` in the blank to print [1, 5, 7]? (Choose all that apply)

```
6: public record Playgroup(String name, int bookings) {
7:     @Override public String toString() { return "" + bookings; }
8: }
9: public static void main(String[] args) {
10:     Playgroup p1 = new Playgroup("Name", 1);
11:     Playgroup p2 = new Playgroup("Name", 2);
12:     Playgroup p3 = new Playgroup("Name", 3);
13:     List<Playgroup> list = Arrays.asList(p1, p2, p3);
14:     Collections.sort(list, Comparator.comparing______);
15:     System.out.println(list);
16: }
```

7. F,B

7. Which of the following method signatures are valid overrides of the `sortBy()` method in the `Algebra` class? (Choose all that apply)

```
7: import java.util.*;
8: public class Algebra {
9:     public static String sortBy(List<String> list) { return null; }
10: }
```

8. B->E

8. Which of the following IDs in the blank, allowing the code to compile and run without issue?

```
8: Sequence<Collection<String>> animals = new _____();
9: animals.addFirst("lion");
10: animals.addLast("tiger");
11: animals.add("elephant");
12: for (var x : animals)
13:     System.out.println(x);
14: System.out.println(animals.get(#));
```

A. HashSet

B. LinkedHashMap

C. TreeSetMap

D. HashMap

E. None of the above

9. A

9. What is the result of the following program?

Because b is before a, is in a reversed order.

```
9: public class MyComparator implements Comparator<String> {
10:     public int compare(String a, String b) {
11:         return String.toLowerCase(a).compareTo(b);
12:     }
13: }
14: public static void main(String[] args) {
15:     String[] values = { "J2EE", "Java", "Web" };
16:     Arrays.sort(values, new MyComparator());
17:     for (var v : values)
18:         System.out.print(v + " ");
19: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

10. A,B,D

10. Which of these statements can fit in the blank so that the `Myelper` class compiles successfully? (Choose all that apply)

```
10: public class Myelper {
11:     public static ? extends Exception
12:         void printException(@) {
13:             System.out.println(e.getMessage());
14:         }
15:     public static void main(String[] args) {
16:         Myelper_____;
17:     }
18: }
```

A. printException(new IllegalStateException("A"))

B. printException(new Exception("B"))

C. printException(new RuntimeException("C"))

2. G, C

3. Which of the following are true? (Choose all that apply)

```
3: List<String> s = list.of("mouse", "java");
4: var x = s.list.of("mouse", "java");
5:
6: s.removal(0);
7: s.removalIf(x.length() == 4);
8: s.removeIf(x -> x.length() == 4);
9: s.removeIf(x -> x.length() == 4);
```

A. This code compiles and runs without error.

B. Exactly one of these lines contains a compiler error.

C. Exactly three of these lines contain a compiler error.

D. Exactly four of these lines contain a compiler error.

E. If any lines with compiler errors are removed, this code runs without an runtime exception.

F. If all lines with compiler errors are removed, this code throws an exception.

3. b

3. What is the result of the following statement?

```
3: var greetings = new Arraylist<String>();
4: greetings.addFirst("Hello");
5: greetings.addLast("Hi");
6: greetings.addFirst("Hello");
7: greetings.addLast("Hi");
8: greetings.pop();
9: while (greetings.pop() != null)
10:     System.out.println(greetings.pop());
```

A. Hello

B. HelloHi

C. HelloHelloHi

D. Hi

E. The code does not compile.

F. An exception is thrown.

4. B,F

4. Which of these statements compile? (Choose all that apply)

- A. HashSet<Character> hs = new HashSet<Character>();
- B. HashSet<? super Comparable> set = new HashSet<Exception>();
- C. List<> list = new ArrayList<String>();
- D. List<Objects> values = new HashSet<Object>();
- E. List<Objects> objects = new ArrayList<? extends Object>();
- F. Map<String, ? extends Number> hm = new HashMap<String, Integer>();

5. E => B

5. What is the result of the following code?

```
5: public record HelloId(int id) {
6:     public HelloId() { this.id = 1; }
7:     public void print(String message) {
8:         System.out.println(id + " " + message);
9:     }
10: }
11: public static void main(String[] args) {
12:     new HelloId().print("Hello");
13:     new HelloId().print("World");
14: }
```

A. hi followed by a runtime exception.

B. hi-hello-world

C. The first compiler error is on line 1.

D. The second compiler error is on line 5.

E. The first compiler error is on the 6.

F. The first compiler error is on another line.

6. B,F

6. Which of the following use `Set` in the blank to print [1, 5, 7]? (Choose all that apply)

```
6: public record Playgroup(String name, int bookings) {
7:     @Override public String toString() { return "" + bookings; }
8: }
9: public static void main(String[] args) {
10:     Playgroup p1 = new Playgroup("Name", 1);
11:     Playgroup p2 = new Playgroup("Name", 2);
12:     Playgroup p3 = new Playgroup("Name", 3);
13:     List<Playgroup> list = Arrays.asList(p1, p2, p3);
14:     Collections.sort(list, Comparator.comparing______);
15:     System.out.println(list);
16: }
```

A. HashSet

B. LinkedHashMap

C. TreeSetMap

D. HashMap

E. None of the above

7. F,B

7. Which of the following method signatures are valid overrides of the `sortBy()` method in the `Algebra` class? (Choose all that apply)

```
7: import java.util.*;
8: public class Algebra {
9:     public static String sortBy(List<String> list) { return null; }
10: }
```

8. B->E

8. Which of the following IDs in the blank, allowing the code to compile and run without issue?

```
8: Sequence<Collection<String>> animals = new _____();
9: animals.addFirst("lion");
10: animals.addLast("tiger");
11: animals.add("elephant");
12: for (var x : animals)
13:     System.out.println(x);
14: System.out.println(animals.get(#));
```

A. HashSet

B. LinkedHashMap

C. TreeSetMap

D. HashMap

E. None of the above

9. A

9. What is the result of the following program?

Because b is before a, is in a reversed order.

```
9: public class MyComparator implements Comparator<String> {
10:     public int compare(String a, String b) {
11:         return String.toLowerCase(a).compareTo(b);
12:     }
13: }
14: public static void main(String[] args) {
15:     String[] values = { "J2EE", "Java", "Web" };
16:     Arrays.sort(values, new MyComparator());
17:     for (var v : values)
18:         System.out.print(v + " ");
19: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

10. A,B,D

10. Which of these statements can fit in the blank so that the `Myelper` class compiles successfully? (Choose all that apply)

```
10: public class Myelper {
11:     public static ? extends Exception
12:         void printException(@) {
13:             System.out.println(e.getMessage());
14:         }
15:     public static void main(String[] args) {
16:         Myelper_____;
17:     }
18: }
```

A. printException(new IllegalStateException("A"))

B. printException(new Exception("B"))

C. printException(new RuntimeException("C"))

D. printException(new Error("D"))

E. None of the above

11. A

11. What is the result of the following program?

Because b is before a, is in a reversed order.

```
11: public class Myelper implements Comparator<Strings> {
12:     public int compare(Strings a, Strings b) {
13:         return String.toLowerCase(a).compareTo(b);
14:     }
15: }
16: public static void main(Strings[] args) {
17:     Strings[] values = { "J2EE", "Java", "Web" };
18:     Arrays.sort(values, new Myelper());
19:     for (var v : values)
20:         System.out.print(v + " ");
21: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

12. A,B,D

12. Which of these statements can fill in the blank so that the `Myelper` class compiles successfully? (Choose all that apply)

```
12: public class Myelper {
13:     public static ? extends Exception
14:         void printException(@) {
15:             System.out.println(e.getMessage());
16:         }
17:     public static void main(@String[] args) {
18:         Myelper_____;
19:     }
20: }
```

A. printException(new IllegalStateException("A"))

B. printException(new Exception("B"))

C. printException(new RuntimeException("C"))

D. printException(new Error("D"))

E. None of the above

13. A

13. What is the result of the following program?

Because b is before a, is in a reversed order.

```
13: public class Myelper {
14:     public static void main(@String[] args) {
15:         Myelper_____;
16:     }
17: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

14. A

14. What is the result of the following program?

Because b is before a, is in a reversed order.

```
14: public class Myelper {
15:     public static void main(@String[] args) {
16:         Myelper_____;
17:     }
18: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

15. A

15. What is the result of the following program?

Because b is before a, is in a reversed order.

```
15: public class Myelper {
16:     public static void main(@String[] args) {
17:         Myelper_____;
18:     }
19: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

16. A

16. What is the result of the following program?

Because b is before a, is in a reversed order.

```
16: public class Myelper {
17:     public static void main(@String[] args) {
18:         Myelper_____;
19:     }
20: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

17. A

17. What is the result of the following program?

Because b is before a, is in a reversed order.

```
17: public class Myelper {
18:     public static void main(@String[] args) {
19:         Myelper_____;
20:     }
21: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

18. A

18. What is the result of the following program?

Because b is before a, is in a reversed order.

```
18: public class Myelper {
19:     public static void main(@String[] args) {
20:         Myelper_____;
21:     }
22: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

19. A

19. What is the result of the following program?

Because b is before a, is in a reversed order.

```
19: public class Myelper {
20:     public static void main(@String[] args) {
21:         Myelper_____;
22:     }
23: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2EE

D. J2EE Web Java

E. The code does not compile.

F. A runtime exception is thrown.

20. A

20. What is the result of the following program?

Because b is before a, is in a reversed order.

```
20: public class Myelper {
21:     public static void main(@String[] args) {
22:         Myelper_____;
23:     }
24: }
```

A. J2EE Java Web

B. Java J2EE Web

C. Java Web J2

Quick Notes Page

19. What code change is needed to make the method compile, assuming there is no class named `X`?

```
public static T newInstance(T x) {  
    return x;  
}
```

- A. Add `x` after the public keyword.
- B. Add `x` after the static keyword.
- C. Add `x` after `T`.
- D. Add `x` after the public keyword.
- E. No change is required, the code already compiles.

20. F, D, F

22. **F, B**

23. What is the output of the following code snippet?

```
21. Selections<Map<Integer, String> cats = new TreeMap<();  
22. cats.put(1, "Sage");  
23. cats.put(2, "Sage");  
24. cats.put(1, "Missis Mouse");  
25. System.out.println(cats);  
26. var id = cats.lastEntry().getKey();  
27. cats.pollFirstEntry();  
28. System.out.println(cats.firstEntry().getvalue());
```

22. B. The code compiles and runs without issue, as options D and E are incorrect. A `TreeMap` sorts its items in the natural order of keys (not the values). Therefore, lines 23 and 27 remove `{1, Missis Mouse}` and `{2, Sage}`, respectively. Line 24 has no impact on the map. On line 28, `Seedwall` is printed, making option B correct. If line 26 were changed to use `pollLastEntry()`, then the map would be empty and line 28 would throw an `IllegalStateException` trying to call `getvalue()`.

- A. Missis Mouse
- B. Seedwall
- C. Sage

D. The code does not compile.

E. The code compiles, but an exception is thrown at runtime.

23. **A, H**

23. What is the output of the following code snippet?

```
24. Fileset x = new TreeSet<String>;  
x.add("one");  
x.add("two");  
x.add("three");  
forvar f1 : Fileset  
System.out.println(f1 + ",");
```

23. H. `TreeSet` is a `SequentialSet`, so it does have an `addFirst()` method. For this reason, the code does compile. Unfortunately, `addFirst()` is not supported at runtime, as inserting an element at the head of the TreeSet contradicts the concept of the `TreeSet`. For this reason, the code program throws an `UnsupportedOperationException` on the third line.

- A. var, close, file,
- B. var, not, close,
- C. close, var, file,
- D. close, file, var,
- E. not, var, close,
- F. not, close, var,
- G. The code does not compile.
- H. The code compiles but throws an exception at runtime.