declared in any code, and it takes two parameters. A Comparator is often implemented using a lambda.

Generics are type parameters for code. To create a class with a generic parameter, add <T> after the class name. You can use any name you want for the type parameter. Single uppercase letters are common choices. Generics allow you to specify wildcards. <?> is an unbounded wildcard that means any type. <? extends Object> is an upper bound that means any type that is Object or extends it. <? extends MyInterface> means any type that implements MyInterface. <? super Number> is a lower bound that means any type that is Number or a superclass. A compiler error results from code that attempts to add an item in a list with an unbounded or upper-bounded wildcard.

Exam Essentials

Pick the correct type collection from a description. A List allows duplicates and orders the elements. A Set does not allow duplicates. A Deque orders its elements to facilitate retrievals from the front or back. A Map maps keys to values. Be familiar with the differences in implementations of these interfaces.

Work with convenience methods. The Collections Framework contains many methods such as contains(), forEach(), and removeIf() that you need to know for the exam. There are too many to list in this paragraph for review, so please do review the tables in this chapter.

Differentiate between Comparable and Comparator. Classes that implement Comparable are said to have a natural ordering and implement the compareTo() method. A class is allowed to have only one natural ordering. A Comparator takes two objects in the compare() method. Different ones can have different sort orders. A Comparator is often implemented using a lambda such as (a, b) -> a.num - b.num.

Identify valid and invalid uses of generics and wildcards. <T> represents a type parameter. Any name can be used, but a single uppercase letter is the convention. <?> is an unbounded wildcard. <? extends X> is an upper-bounded wildcard. <? super X> is a lower-bounded wildcard.

Review Questions

The answers to the chapter review questions can be found in the Appendix.

1. Suppose you need to display a collection of products for sale, which may contain duplicates. Additionally, you have a collection of sales that you need to track, sorted by the natural order of the sale ID, and you need to retrieve the text of each. Which two of the following from the java.util package best suit your needs for this scenario? (Choose two.)

A. ArrayList

B. HashMap

C. HashSet

D. LinkedList

E. TreeMap

F. TreeSet

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

```
12: List<?> q = List.of("mouse", "parrot");
13: var v = List.of("mouse", "parrot");
14:
15: q.removeIf(String::isEmpty);
16: q.removeIf(s -> s.length() == 4);
17: v.removeIf(String::isEmpty);
18: v.removeIf(s -> s.length() == 4);
```

A. This code compiles and runs without error.

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

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

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

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

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

G. If any lines with compiler errors are removed, this code throws an exception.

3. What is the result of the following statements?

```
3:  var greetings = new ArrayDeque<String>();
4:  greetings.offerLast("hello");
5:  greetings.offerLast("hi");
6:  greetings.offerFirst("ola");
7:  greetings.pop();
8:  greetings.peek();
9:  while (greetings.peek() != null)
10:   System.out.print(greetings.pop());
```

A. hello

B. hellohi

C. hellohiola

D. hiola

E. The code does not compile.

F. An exception is thrown.

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

A. HashSet<Number> hs = new HashSet<Integer>();

B. HashSet<? super ClassCastException> set = new HashSet<Exception>();

C. List<> list = new ArrayList<String>();

D. List<Object> values = new HashSet<Object>();

E. List<Object> objects = new ArrayList<? extends Object>();

F. Map<String, ? extends Number> hm = new HashMap<String, Integer>();

5. What is the result of the following code?

```
1: public record Hello<T>(T t) {
2:   public Hello(T t) { this.t = t; }
3:   private <T> void println(T message) {
4:     System.out.print(t + "-" + message);
5:   }
6:   public static void main(String[] args) {
7:     new Hello<String>("hi").println(1);
8:     new Hello("hola").println(true);
9:   } }
```

A. hi followed by a runtime exception

B. hi-1hola-true

C. The first compiler error is on line 1.

D. The first compiler error is on line 3.

E. The first compiler error is on line 8.

F. The first compiler error is on another line.

6. Which of the following can fill in the blank to print [7, 5, 3]? (Choose all that apply.)

```
8:  public record Platypus(String name, int beakLength) {
9:    @Override public String toString() {return "" + beakLength;}
10:
11:   public static void main(String[] args) {
12:     Platypus p1 = new Platypus("Paula", 3);
13:     Platypus p2 = new Platypus("Peter", 5);
14:     Platypus p3 = new Platypus("Peter", 7);
```

```
15:
16:     List<Platypus> list = Arrays.asList(p1, p2, p3);
17:
18:     Collections.sort(list, Comparator.comparing_____);
19:
20:     System.out.println(list);
21:   }
22: }
```

A. (Platypus::beakLength)

B. (Platypus::beakLength).reversed()

C.
```
(Platypus::name)
    .thenComparing(Platypus::beakLength)
```

D.
```
(Platypus::name)
    .thenComparing(
    Comparator.comparing(Platypus::beakLength)
    .reversed())
```

E.
```
(Platypus::name)
    .thenComparingNumber(Platypus::beakLength)
    .reversed()
```

F.
```
(Platypus::name)
    .thenComparingInt(Platypus::beakLength)
    .reversed()
```

G. None of the above

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

```
import java.util.*;

public class Alpaca {
  public List<String> hairy(List<String> list) { return null; }
}
```

A. public List<String> hairy(List<CharSequence> list) { return null; }

B. public List<String> hairy(ArrayList<String> list) { return null; }

C. public List<String> hairy(List<Integer> list) { return null; }

D. public List<CharSequence> hairy(List<String> list) { return null; }

E. public Object hairy(List<String> list) { return null; }

F. public ArrayList<String> hairy(List<String> list) { return null; }

8. What is the result of the following program?

```
3:  public class MyComparator implements Comparator<String> {
4:    public int compare(String a, String b) {
5:      return b.toLowerCase().compareTo(a.toLowerCase());
6:    }
7:    public static void main(String[] args) {
8:      String[] values = { "123", "Abb", "aab" };
9:      Arrays.sort(values, new MyComparator());
10:     for (var s: values)
11:       System.out.print(s + " ");
12:   }
13: }
```

A. Abb aab 123

B. aab Abb 123

C. 123 Abb aab

D. 123 aab Abb

E. The code does not compile.

F. A runtime exception is thrown.

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

```
2:  public class Helper {
3:    public static <U extends Exception>
4:      void printException(U u) {
5:
6:      System.out.println(u.getMessage());
7:    }
8:    public static void main(String[] args) {
9:      Helper._____;
10:   } }
```

A. printException(new FileNotFoundException("A"))

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

C. <Throwable>printException(new Exception("C"))

D. <NullPointerException>printException(new NullPointerException ("D"))

E. printException(new Throwable("E"))

10. Which of the following will compile when filling in the blank? (Choose all that apply.)

```
var list = List.of(1, 2, 3);
var set = Set.of(1, 2, 3);
var map = Map.of(1, 2, 3, 4);

_____.forEach(System.out::println);
```

A. list

B. set

C. map

D. map.keys()

E. map.keySet()

F. map.values()

G. map.valueSet()

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

```
3:  public class Wildcard {
4:    public void showSize(List<?> list) {
5:      System.out.println(list.size());
6:    }
7:    public static void main(String[] args) {
8:      Wildcard card = new Wildcard();
9:      _____;
10:     card.showSize(list);
11:   }}
```

A. List<?> list = new HashSet <String>()

B. ArrayList<? super Date> list = new ArrayList<Date>()

C. List<?> list = new ArrayList<?>()

D. List<Exception> list = new LinkedList<java.io.IOException>()

E. ArrayList <? extends Number> list = new ArrayList <Integer>()

F. None of the above

12. What is the result of the following program?

```
3:  public record Sorted(int num, String text)
4:    implements Comparable<Sorted>, Comparator<Sorted> {
5:
6:    public String toString() { return "" + num; }
7:    public int compareTo(Sorted s) {
8:      return text.compareTo(s.text);
9:    }
10:   public int compare(Sorted s1, Sorted s2) {
11:     return s1.num - s2.num;
12:   }
13:   public static void main(String[] args) {
14:     var s1 = new Sorted(88, "a");
15:     var s2 = new Sorted(55, "b");
16:     var t1 = new TreeSet<Sorted>();
17:     t1.add(s1); t1.add(s2);
18:     var t2 = new TreeSet<Sorted>(s1);
19:     t2.add(s1); t2.add(s2);
20:     System.out.println(t1 + " " + t2);
21:   } }
```

A. [55, 88] [55, 88]

B. [55, 88] [88, 55]

C. [88, 55] [55, 88]

D. [88, 55] [88, 55]

E. The code does not compile.

F. A runtime exception is thrown.

13. What is the result of the following code? (Choose all that apply.)

```
Comparator<Integer> c1 = (o1, o2) -> o2 - o1;
Comparator<Integer> c2 = Comparator.naturalOrder();
Comparator<Integer> c3 = Comparator.reverseOrder();

var list = Arrays.asList(5, 4, 7, 2);
Collections.sort(list,_________);
Collections.reverse(list);
Collections.reverse(list);
System.out.println(Collections.binarySearch(list, 2));
```

A. One or more of the comparators can fill in the blank so that the code prints 0.

B. One or more of the comparators can fill in the blank so that the code prints 1.

C. One or more of the comparators can fill in the blank so that the code prints 2.

D. The result is undefined regardless of which comparator is used.

E. A runtime exception is thrown regardless of which comparator is used.

F. The code does not compile.

14. Which of the following lines can be inserted to make the code compile? (Choose all that apply.)

```
class W {}
class X extends W {}
class Y extends X {}
class Z<Y> {
  // INSERT CODE HERE
}
```

A. W w1 = new W();

B. W w2 = new X();

C. W w3 = new Y();

D. Y y1 = new W();

E. Y y2 = new X();

F. Y y1 = new Y();

15. Which options are true of the following code? (Choose all that apply.)

```
3:_____ q = new LinkedList<>();
4: q.add(10);
5: q.add(12);
6: q.remove(1);
7: System.out.print(q);
```

A. If we fill in the blank with List<Integer>, the output is [10].

B. If we fill in the blank with Queue<Integer>, the output is [10].

C. If we fill in the blank with var, the output is [10].

D. One or more of the scenarios does not compile.

E. One or more of the scenarios throws a runtime exception.

16. What is the result of the following code?

```
4: Map m = new HashMap();
5: m.put(123, "456");
6: m.put("abc", "def");
7: System.out.println(m.contains("123"));
```

A. false

B. true

C. Compiler error on line 4

D. Compiler error on line 5

E. Compiler error on line 7

F. A runtime exception is thrown.

17. What is the result of the following code? (Choose all that apply.)

```
48: var map = Map.of(1,2, 3, 6);
49: var list = List.copyOf(map.entrySet());
50:
51: List<Integer> one = List.of(8, 16, 2);
52: var copy = List.copyOf(one);
53: var copyOfCopy = List.copyOf(copy);
54: var thirdCopy = new ArrayList<>(copyOfCopy);
55:
56: list.replaceAll(x -> x * 2);
57: one.replaceAll(x -> x * 2);
58: thirdCopy.replaceAll(x -> x * 2);
59:
60: System.out.println(thirdCopy);
```

A. One line fails to compile.

B. Two lines fail to compile.

C. Three lines fail to compile.

D. The code compiles but throws an exception at runtime.

E. If any lines with compiler errors are removed, the code throws an exception at runtime.

F. If any lines with compiler errors are removed, the code prints [16, 32, 4].

G. The code compiles and prints [16, 32, 4] without any changes.

18. What code change is needed to make the method compile, assuming there is no class named T?

```
public static T identity(T t) {
  return t;
}
```

A. Add <T> after the public keyword.

B. Add <T> after the static keyword.

C. Add <T> after T.

D. Add <?> after the public keyword.

E. Add <?> after the static keyword.

F. No change is required. The code already compiles.

19. What is the result of the following?

```
var map = new HashMap<Integer, Integer>();
map.put(1, 10);
map.put(2, 20);
map.put(3, null);
map.merge(1, 3, (a,b) -> a + b);
map.merge(3, 3, (a,b) -> a + b);
System.out.println(map);
```

A. {1=10, 2=20}

B. {1=10, 2=20, 3=null}

C. {1=10, 2=20, 3=3}

D. {1=13, 2=20}

E. {1=13, 2=20, 3=null}

F. {1=13, 2=20, 3=3}

G. The code does not compile.

H. An exception is thrown.

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

A. Comparable is in the java.util package.

B. Comparator is in the java.util package.

C. compare() is in the Comparable interface.

D. compare() is in the Comparator interface.

E. compare() takes one method parameter.

F. compare() takes two method parameters.