

<b>Started on</b>	Thursday, 5 June 2025, 11:31 AM
<b>State</b>	Finished
<b>Completed on</b>	Thursday, 5 June 2025, 12:09 PM
<b>Time taken</b>	37 mins 17 secs
<b>Marks</b>	31.00/50.00
<b>Grade</b>	<b>62.00</b> out of 100.00

**Question 1**

Complete

Mark 0.00 out of  
1.00

What will be the output of the following code?

```
class A {
    int x = 10;
    A() {
        print();
    }
    void print() {
        System.out.println("A: " + x);
    }
}
class B extends A {
    int x = 20;
    void print() {
        System.out.println("B: " + x);
    }
}
public class Main {
    public static void main(String[] args) {
        A obj = new B();
    }
}
```

- ☐ a. B: 20
- ☐ b. Runtime Error
- ☒ c. A: 10
- ☐ d. B: 0

**Question 2**

Complete

Mark 0.00 out of  
1.00

What is the result of the following code?

```
interface I1 {
    default void display() {
        System.out.println("I1");
    }
}

interface I2 {
    default void display() {
        System.out.println("I2");
    }
}

class C implements I1, I2 {
    public void display() {
        I1.super.display();
    }
}

public class Test {
    public static void main(String[] args) {
        new C().display();
    }
}
```

- ☒ a. Compilation error
- ☐ b. I1
- ☐ c. I2
- ☐ d. I1 I2

**Question 3**

Complete

Mark 0.00 out of  
1.00

What will this code print?

```
class Super {
    static void method() {
        System.out.println("Super");
    }
}
class Sub extends Super {
    static void method() {
        System.out.println("Sub");
    }
}
public class Demo {
    public static void main(String[] args) {
        Super obj = new Sub();
        obj.method();
    }
}
```

- ☐ a. Super
- ☐ b. Sub
- ☐ c. Runtime error
- ☒ d. Compilation error

**Question 4**

Complete

Mark 1.00 out of  
1.00

Which one is not allowed in Java?

- ☐ a. Abstract class with constructor
- ☐ b. Abstract class with static methods
- ☐ c. A class with both abstract and non-abstract methods
- ☒ d. Final class with abstract methods

**Question 5**

Complete

Mark 0.00 out of  
1.00

What is the output?

```
class Test {  
    private void display() {  
        System.out.println("Private");  
    }  
}  
class Sub extends Test {  
    public void display() {  
        System.out.println("Public");  
    }  
}  
public class Demo {  
    public static void main(String[] args) {  
        Test t = new Sub();  
        t.display();  
    }  
}
```

- ☐ a. Private
- ☐ b. **Compilation error**
- ☐ c. Runtime error
- ☒ d. Public

**Question 6**

Complete

Mark 1.00 out of 1.00

What is the output of the following code?

```
class Base {  
    void show() {  
        System.out.println("Base show()");  
    }  
}  
class Derived extends Base {  
    void show(int x) {  
        System.out.println("Derived show(" + x + ")");  
    }  
}  
public class Main {  
    public static void main(String[] args) {  
        Base b = new Derived();  
        b.show();  
    }  
}
```

- ☐ a. Compilation error
- ☒ b. Base show()
- ☐ c. Derived show()
- ☐ d. Runtime error

**Question 7**

Complete

Mark 1.00 out of 1.00

What is the output?

```
class A {
    int i = 10;
    A() {
        System.out.println(i);
        i = 20;
    }
}

public class Test {
    public static void main(String[] args) {
        A a = new A();
        System.out.println(a.i);
    }
}
```

- ☐ a. 10 10
- ☐ b. 20 20
- ☒ c. 10 20
- ☐ d. 0 10

**Question 8**

Complete

Mark 0.00 out of 1.00

What is true about constructors in Java?

- ☐ a. A constructor can call another constructor in the same class using this()
- ☒ b. Constructors can be inherited
- ☐ c. Constructors must be public
- ☐ d. Constructor name can be different from class name

**Question 9**

Complete

Mark 1.00 out of 1.00

What will be the output?

```
class Animal {  
    void sound() { System.out.println("Generic sound"); }  
}  
class Dog extends Animal {  
    void sound() { System.out.println("Bark"); }  
}  
public class Main {  
    public static void main(String[] args) {  
        Animal a = new Dog();  
        a.sound();  
    }  
}
```

- ☐ a. Runtime error
- ☐ b. Generic sound
- ☐ c. Compilation error
- ☒ d. Bark

**Question 10**

Complete

Mark 1.00 out of 1.00

Which of the following is not true for method overriding?

- ☐ a. Method name must be same
- ☐ b. Return type must be same or subtype
- ☒ c. Access modifier can be more restrictive
- ☐ d. Method must be inherited



**Question 11**

Complete

Mark 1.00 out of 1.00

What will happen?

```
class A {  
    final void show() {}  
}  
class B extends A {  
    void show() {}  
}
```

- ☐ a. Runtime error
- ☐ b. Shows nothing
- ☒ c. Compilation error
- ☐ d. Compiles and runs

**Question 12**

Complete

Mark 1.00 out of 1.00

What is the output?

```
try {  
    throw new IllegalArgumentException("Illegal");  
} catch (IllegalArgumentException e) {  
    System.out.println(e.getMessage());  
}
```

- ☐ a. Exception
- ☐ b. IllegalArgumentException
- ☒ c. Illegal
- ☐ d. null

**Question 13**

Complete

Mark 1.00 out of 1.00

What is true about finally block?

- ☒ a. It always executes regardless of exceptions
- ☐ b. It executes only if no exception occurs
- ☐ c. It is optional
- ☐ d. It must be used with catch

**Question 14**

Complete

Mark 1.00 out of 1.00

What will be printed?

```
try {  
    int[] arr = new int[5];  
    arr[5] = 100;  
} catch (ArrayIndexOutOfBoundsException e) {  
    System.out.print("Caught ");  
}  
finally {  
    System.out.print("Finally");  
}
```

- ☐ a. Finally
- ☐ b. Runtime Error
- ☐ c. Caught
- ☒ d. Caught Finally

**Question 15**

Complete

Mark 1.00 out of 1.00

What happens if catch block is missing?

```
try {  
    System.out.println(10 / 0);  
} finally {  
    System.out.println("Done");  
}
```

- ☐ a. Done
- ☒ b. ArithmeticException is thrown after finally
- ☐ c. Program compiles but doesn't run
- ☐ d. Compilation error

**Question 16**

Complete

Mark 0.00 out of 1.00

What is the output of the following code?

```
try {  
    throw new Exception("Check");  
} catch (RuntimeException e) {  
    System.out.println("Runtime");  
} catch (Exception e) {  
    System.out.println("Exception");  
}
```

- ☒ a. Exception
- ☐ b. Compilation error
- ☐ c. Runtime
- ☐ d. Runtime error

**Question 17**

Complete

Mark 0.00 out of 1.00

Which of the following is a checked exception?

- ☐ a. FileNotFoundException
- ☐ b. NullPointerException
- ☒ c. IllegalArgumentException
- ☐ d. ArithmeticException

**Question 18**

Complete

Mark 1.00 out of 1.00

What is the output?

```
try {  
    int a = 5 / 0;  
} catch (Exception e) {  
    System.out.print("Catch ");  
} finally {  
    System.out.print("Finally");  
}
```

- ☐ a. Runtime Error
- ☒ b. Catch Finally
- ☐ c. Catch
- ☐ d. Finally

**Question 19**

Complete

Mark 0.00 out of 1.00

What happens if exception is thrown in finally block?

- ☐ a. Original exception is suppressed
- ☐ b. finally block never throws exception
- ☐ c. Compile-time error
- ☒ d. Both exceptions are printed

**Question 20**

Complete

Mark 0.00 out of 1.00

What is the output?

```
public class Test {  
    public static void main(String[] args) {  
        try {  
            System.out.println("Try");  
            return;  
        } finally {  
            System.out.println("Finally");  
        }  
    }  
}
```

- ☐ a. Try Finally
- ☒ b. Try
- ☐ c. Finally
- ☐ d. Compilation error

**Question 21**

Complete

Mark 0.00 out of 1.00

What is the output of the following code?

```
LocalDate date = LocalDate.of(2024, Month.FEBRUARY, 29);  
System.out.println(date.plusYears(1));
```

- ☐ a. 2025-02-29
- ☒ b. 2025-03-01
- ☐ c. 2025-02-28
- ☐ d. DateTimeException

**Question 22**

Complete

Mark 1.00 out of 1.00

What does this print?

```
LocalTime time = LocalTime.of(23, 59, 59);  
System.out.println(time.plusSeconds(2));
```

- ☒ a. 00:00:01
- ☐ b. 00:01:00
- ☐ c. 23:59:61
- ☐ d. 00:00

**Question 23**

Complete

Mark 1.00 out of 1.00

Which class would you use to represent a date and time with time zone?

- ☐ a. LocalDateTime
- ☐ b. OffsetDateTime
- ☒ c. ZonedDateTime
- ☐ d. Instant

**Question 24**

Complete

Mark 0.00 out of 1.00

What will this code output?

```
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy/MM/dd");  
LocalDate date = LocalDate.parse("2023/08/15", formatter);  
System.out.println(date);
```

- ☐ a. 2023-08-15
- ☐ b. 2023/08/15
- ☐ c. Compilation Error
- ☒ d. August 15, 2023

**Question 25**

Complete

Mark 0.00 out of 1.00

What is the output?

```
Instant instant = Instant.now();  
ZonedDateTime zdt = instant.atZone(ZoneId.of("UTC"));  
System.out.println(zdt.getOffset());
```

- ☐ a. Throws Exception
- ☐ b. null
- ☒ c. System timezone offset
- ☐ d. +00:00

**Question 26**

Complete

Mark 1.00 out of 1.00

What does the JVM use the heap memory for?

- ☐ a. Stack frames
- ☐ b. Thread-local storage
- ☐ c. Method calls
- ☒ d. Object allocation

**Question 27**

Complete

Mark 1.00 out of 1.00

What happens when an object becomes unreachable in Java?

- ☐ a. It's immediately deleted
- ☐ b. JVM throws NullPointerException
- ☐ c. StackOverflowError occurs
- ☒ d. It becomes eligible for GC

**Question 28**

Complete

Mark 0.00 out of 1.00

Which generation in GC typically contains short-lived objects?

- ☐ a. Eden Space
- ☒ b. Tenured
- ☐ c. Old Generation
- ☐ d. PermGen

**Question 29**

Complete

Mark 0.00 out of 1.00

What is true about finalize() in Java?

- ☐ a. It may or may not be called
- ☒ b. It is always called before GC
- ☐ c. It is used for performance
- ☐ d. It guarantees memory cleanup



**Question 30**

Complete

Mark 0.00 out of 1.00

What is the role of System.gc()?

- ☐ a. Prevents GC
- ☒ b. Immediately triggers GC
- ☐ c. Deletes static data
- ☐ d. Requests GC but does not force it

**Question 31**

Complete

Mark 0.00 out of 1.00

What is the output?

```
public class Test extends Thread {  
    public void run() {  
        System.out.println("Thread running");  
    }  
    public static void main(String[] args) {  
        Test t = new Test();  
        t.run();  
        t.start();  
    }  
}
```

- ☐ a. Compilation error
- ☐ b. Runtime error
- ☒ c. Only one "Thread running"
- ☐ d. Thread running Thread running

**Question 32**

Complete

Mark 0.00 out of 1.00

Which method causes the current thread to wait until another completes?

- ☐ a. yield()
- ☒ b. wait()
- ☐ c. join()
- ☐ d. sleep()

**Question 33**

Complete

Mark 1.00 out of 1.00

What will this code output?

```
public class Main {  
    public static void main(String[] args) {  
        Thread t = new Thread(() -> {  
            for (int i = 0; i < 2; i++) {  
                System.out.print(Thread.currentThread().getName() + " ");  
            }  
        });  
        t.setName("Worker");  
        t.start();  
    }  
}
```

- ☐ a. Compiler Error
- ☒ b. Worker Worker
- ☐ c. Main Main
- ☐ d. Thread-0 Thread-0

**Question 34**

Complete

Mark 1.00 out of 1.00

What does this print?

```
class MyThread extends Thread {  
    public void run() {  
        System.out.print("Hello ");  
    }  
    public static void main(String[] args) {  
        MyThread t1 = new MyThread();  
        t1.start();  
        t1.start();  
    }  
}
```

- ☒ a. Runtime Exception
- ☐ b. Compilation Error
- ☐ c. Hello Hello
- ☐ d. Hello

**Question 35**

Complete

Mark 0.00 out of 1.00

Which of the following is true about thread priorities?

- ☐ a. Priorities are always respected by JVM
- ☐ b. Higher priority means more CPU time guaranteed
- ☒ c. Priority affects the order of execution deterministically
- ☐ d. Thread priority is a hint to the thread scheduler

**Question 36**

Complete

Mark 1.00 out of 1.00

What does this code print?

```
import java.util.concurrent.*;
public class Main {
    public static void main(String[] args) {
        ExecutorService service = Executors.newFixedThreadPool(2);
        service.submit(() -> System.out.println("Task 1"));
        service.submit(() -> System.out.println("Task 2"));
        service.shutdown();
    }
}
```

- ☐ a. Compilation error
- ☐ b. Task 2 Task 1
- ☐ c. Task 1 Task 2
- ☒ d. Any order of Task 1 and Task 2

**Question 37**

Complete

Mark 0.00 out of 1.00

What is the key feature of ReentrantLock over synchronized?

- ☐ a. Simpler syntax
- ☒ b. No need to unlock manually
- ☐ c. Can't be interrupted
- ☐ d. Fairness policy and tryLock() capability

**Question 38**

Complete

Mark 1.00 out of 1.00

What will happen here?

```
ConcurrentHashMap<String, Integer> map = new ConcurrentHashMap<>();  
map.put("A", 1);  
map.compute("A", (k, v) -> v + 1);  
System.out.println(map.get("A"));
```

- ☐ a. 1
- ☐ b. Compilation error
- ☒ c. 2
- ☐ d. NullPointerException

**Question 39**

Complete

Mark 1.00 out of 1.00

What is ForkJoinPool best suited for?

- ☐ a. Long I/O-bound tasks
- ☒ b. Dividing tasks recursively and processing in parallel
- ☐ c. Thread communication
- ☐ d. UI event handling

**Question 40**

Complete

Mark 1.00 out of 1.00

What is the default parallelism level of ForkJoinPool?

- ☐ a. 1
- ☒ b. Number of available processors
- ☐ c. Fixed at 4
- ☐ d. Number of processors - 1

**Question 41**

Complete

Mark 1.00 out of 1.00

What will this code print?

```
List<String> list = Arrays.asList("apple", "banana", "cherry");  
list.stream()  
    .filter(s -> s.length() > 5)  
    .map(String::toUpperCase)  
    .forEach(System.out::print);
```

- ☐ a. APPLEBANANACHERRY
- ☒ b. BANANACHERRY
- ☐ c. banana cherry
- ☐ d. CHERRYBANANA

**Question 42**

Complete

Mark 1.00 out of 1.00

What is the output of the following?

```
Stream.of(1, 2, 3, 4, 5)  
    .filter(i -> i % 2 == 0)  
    .map(i -> i * i)  
    .findFirst()  
    .ifPresent(System.out::print);
```

- ☒ a. 4
- ☐ b. 16
- ☐ c. 1
- ☐ d. 2

**Question 43**

Complete

Mark 1.00 out of 1.00

Which operation is terminal in streams?

- ☐ a. peek()
- ☐ b. map()
- ☒ c. forEach()
- ☐ d. filter()

**Question 44**

Complete

Mark 1.00 out of 1.00

What is the result?

```
List<String> list = Arrays.asList("a", "bb", "ccc", "dd");  
String result = list.stream()  
    .filter(s -> s.length() == 2)  
    .collect(Collectors.joining("-"));  
System.out.println(result);
```

- ☒ a. bb-dd
- ☐ b. bbd
- ☐ c. a-bb-ccc-dd
- ☐ d. bb

**Question 45**

Complete

Mark 1.00 out of 1.00

What will this output?

```
Stream<String> s = Stream.of("java", "lambda", "stream");  
long count = s.map(String::length).filter(1 -> 1 > 5).count();  
System.out.println(count);
```

- ☐ a. 0
- ☒ b. 2
- ☐ c. 1
- ☐ d. 3

**Question 46**

Complete

Mark 1.00 out of 1.00

What is the output?

```
List<String> list = new ArrayList<>();  
list.add("a");  
list.add("b");  
list.add(1, "c");  
System.out.println(list);
```

- ☐ a. Compilation error
- ☐ b. [a, b, c]
- ☐ c. [c, a, b]
- ☒ d. [a, c, b]



**Question 47**

Complete

Mark 1.00 out of 1.00

Which collection is synchronized?

- ☐ a. HashMap
- ☒ b. Vector
- ☐ c. ArrayList
- ☐ d. LinkedList

**Question 48**

Complete

Mark 1.00 out of 1.00

What does this print?

```
Map<String, String> map = new HashMap<>();  
map.put("a", "apple");  
map.put("b", "banana");  
map.put("a", "avocado");  
System.out.println(map.get("a"));
```

- ☒ a. avocado
- ☐ b. banana
- ☐ c. null
- ☐ d. apple

**Question 49**

Complete

Mark 1.00 out of 1.00

What is true about HashSet?

- ☐ a. Allows duplicate elements
- ☒ b. Uses hashCode and equals
- ☐ c. Maintains insertion order
- ☐ d. Implements List

**Question 50**

Complete

Mark 1.00 out of 1.00

Which Map maintains insertion order?

- ☐ a. TreeMap
- ☐ b. HashMap
- ☐ c. Hashtable
- ☒ d. LinkedHashMap