

Started on	Monday, 9 June 2025, 4:27 PM
State	Finished
Completed on	Monday, 9 June 2025, 5:00 PM
Time taken	32 mins 41 secs
Marks	37.00/50.00
Grade	74.00 out of 100.00

Question 1

Complete

Mark 0.00 out of 1.00

What happens when an object becomes unreachable in Java?

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

Question 2

Complete

Mark 0.00 out of 1.00

What is true about finalize() in Java?

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

Question 3

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

Question 4

Complete

Mark 1.00 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. Runtime error
- ☐ b. Private
- ☐ c. Public
- ☒ d. Compilation error

Question 5

Complete

Mark 1.00 out of 1.00

Which of the following is true about thread priorities?

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

Question 6

Complete

Mark 1.00 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 must be public
- ☐ c. Constructors can be inherited
- ☐ d. Constructor name can be different from class name

Question 7

Complete

Mark 0.00 out of 1.00

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

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

Question 8

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 9

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

Question 10

Complete

Mark 1.00 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: 0
- ☐ b. B: 20
- ☐ c. Runtime Error
- ☐ d. A: 10

Question 11

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

Question 12

Complete

Mark 0.00 out of 1.00

Which operation is terminal in streams?

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

Question 13

Complete

Mark 1.00 out of 1.00

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

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

Question 14

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. Fairness policy and `tryLock()` capability
- ☐ d. Can't be interrupted

Question 15

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(l -> l > 5).count();  
System.out.println(count);
```

- ☒ a. 2
- ☐ b. 3
- ☐ c. 0
- ☐ d. 1

Question 16

Complete

Mark 1.00 out of 1.00

Which Map maintains insertion order?

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

Question 17

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. +00:00
- ☐ c. null
- ☒ d. System timezone offset

Question 18

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

Question 19

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

Question 20

Complete

Mark 1.00 out of 1.00

What will happen?

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

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

Question 21

Complete

Mark 1.00 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. Runtime error
- ☒ b. Super
- ☐ c. Compilation error
- ☐ d. Sub

Question 22

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. 16
- ☐ b. 1
- ☐ c. 2
- ☒ d. 4

Question 23

Complete

Mark 1.00 out of 1.00

What is the role of System.gc()?

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

Question 24

Complete

Mark 1.00 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. Compilation error
- ☐ b. Finally
- ☒ c. Try Finally
- ☐ d. Try

Question 25

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

Question 26

Complete

Mark 1.00 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. I1 I2
- ☐ b. I2
- ☒ c. I1
- ☐ d. Compilation error

Question 27

Complete

Mark 1.00 out of 1.00

What is true about HashSet?

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

Question 28

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

Question 29

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

Question 30

Complete

Mark 1.00 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. Compilation Error
- ☐ b. August 15, 2023
- ☐ c. 2023/08/15
- ☒ d. 2023-08-15

Question 31

Complete

Mark 0.00 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. Task 1 Task 2
- ☐ b. Task 2 Task 1
- ☐ c. Compilation error
- ☐ d. Any order of Task 1 and Task 2

Question 32

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

Question 33

Complete

Mark 0.00 out of 1.00

What happens if exception is thrown in finally block?

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

Question 34

Complete

Mark 1.00 out of 1.00

What does the JVM use the heap memory for?

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

Question 35

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

Question 36

Complete

Mark 1.00 out of 1.00

Which collection is synchronized?

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

Question 37

Complete

Mark 0.00 out of 1.00

Which generation in GC typically contains short-lived objects?

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

Question 38

Complete

Mark 1.00 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. Runtime error
- ☐ b. Compilation error
- ☒ c. Exception
- ☐ d. Runtime

Question 39

Complete

Mark 1.00 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. Thread running Thread running
- ☐ b. Only one "Thread running"
- ☐ c. Compilation error
- ☐ d. Runtime error

Question 40

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. Finally
- ☐ c. Catch
- ☒ d. Catch Finally

Question 41

Complete

Mark 0.00 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. Hello
- ☐ b. Compilation Error
- ☒ c. Hello Hello
- ☐ d. Runtime Exception

Question 42

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

Question 43

Complete

Mark 1.00 out of 1.00

What is true about finally block?

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

Question 44

Complete

Mark 0.00 out of 1.00

What happens if catch block is missing?

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

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

Question 45

Complete

Mark 1.00 out of 1.00

Which of the following is a checked exception?

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

Question 46

Complete

Mark 1.00 out of 1.00

Which one is not allowed in Java?

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

Question 47

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

Question 48

Complete

Mark 0.00 out of 1.00

What is ForkJoinPool best suited for?

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

Question 49

Complete

Mark 1.00 out of 1.00

Which of the following is not true for method overriding?

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

Question 50

Complete

Mark 1.00 out of 1.00

What is the default parallelism level of ForkJoinPool?

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