

Time left 0:19:06

Question 1

Not yet answered

Marked out of 1.00

What is the output of the following C++ code?

```
int x = 10;
auto lambda = [=]() mutable { x += 5; return x; };
lambda();
cout << x;
```

- a. 15
- b. 10
- c. Compile error
- d. Undefined behavior

Question 2

Not yet answered

Marked out of 1.00

What does the following code print?

```
struct Base { virtual void f() { cout << "B"; } };
struct Derived : Base { void f() override { cout << "D"; } };
Base b = Derived();
Base b = Derived();
b.f();
```

- a. Compilation error
- b. B
- c. D
- d. Undefined behavior

Question 3

Not yet answered

Marked out of 1.00

Which of the following best describes the Rule of Five?

- a. To avoid using raw pointers in modern C++
- b. Defines how many parameters a template can take
- c. Always use smart pointers for dynamic memory
- d. If you define one of: destructor, copy constructor, copy assignment, move constructor, move assignment — you should define all

Question 4

Not yet answered

Marked out of 1.00

Which of the following leads to a data race?

- a. One thread writing while another thread reads a non-atomic variable
- b. Two threads reading the same variable
- c. Two threads writing to atomic variables
- d. Threads using mutex-protected code

Question 5

Not yet answered

Marked out of 1.00

What happens when two threads call wait() on the same object?

- a. Both threads go to WAITING state until notified
- b. Only one thread becomes BLOCKED
- c. JVM chooses one at random
- d. Both threads wake up immediately

Question 6

Not yet answered

Marked out of 1.00

In C++, which operation is guaranteed not to throw an exception?

- a. noexcept move constructor
- b. Copy construction of a class containing std::string
- c. std::swap() of a vector
- d. std::move() of a vector

Question 7

Not yet answered

Marked out of 1.00

What happens if a std::vector grows beyond its current capacity?

- a. It grows in-place without relocation
- b. It throws an exception
- c. It blocks until another vector releases memory
- d. It reallocates memory and invalidates all pointers/references

Question 8

Not yet answered

Marked out of 1.00

What is the output?

```
String s1 = new String("abc");
String s2 = "abc";
System.out.println(s1 == s2);
```

- a. Compile error
- b. true
- c. Depends on JVM
- d. false

Question 9

Not yet answered

Marked out of 1.00

What will happen in this multithreaded code?

```
synchronized void add() {
    add();
}
```

- a. ReentrantLock will solve it automatically
- b. Compilation error
- c. StackOverflowError
- d. Deadlock

Question 10

Not yet answered

Marked out of 1.00

5.What will the following code print?

- a. 2
- b. 1
- c. 0
- d. Throws NullPointerException

Question 11

Not yet answered

Marked out of 1.00

6. What is the output?

- a. false only for values > 127
- b. false
- c. true
- d. true only for values < -128

Question 12

Not yet answered

Marked out of 1.00

What happens if an exception is thrown during stack unwinding and not handled?

- a. All stack variables stay intact
- b. Program terminates by calling std::terminate()
- c. Program continues after catch block
- d. Causes undefined behavior only in debug mode

Question 13

Not yet answered

Marked out of 1.00

What happens when a subclass defines a method with the same signature as a final method in the parent class?

- a. Compilation error
- b. It overrides it
- c. Runtime exception
- d. It hides it

Question 14

Not yet answered

Marked out of 1.00

What is guaranteed about a class with a virtual destructor?

- a. It must implement all pure virtual methods
- b. It cannot be inherited
- c. It prevents object slicing
- d. Deleting through a base pointer calls the derived destructor

Question 15

Not yet answered

Marked out of 1.00

What is guaranteed by the Java Memory Model for a volatile variable?

- a. Visibility across threads
- b. It prevents all race conditions
- c. It prevents thread starvation
- d. Atomicity for i++ operations

Question 16

Not yet answered

Marked out of 1.00

What is the biggest difference between unique_ptr and shared_ptr?

- a. Both have the same ownership semantics
- b. shared_ptr is faster than unique_ptr
- c. shared_ptr manages ownership through reference counting
- d. unique_ptr supports custom deleters, shared_ptr does not

Question 17

Not yet answered

Marked out of 1.00

Which C++ feature allows compile-time computation?

- a. RTTI
- b. constexpr
- c. inline
- d. lambda functions

Question 18

Not yet answered

Marked out of 1.00

Which class is NOT thread-safe?

- a. ConcurrentHashMap
- b. StringBuilder
- c. CopyOnWriteArrayList
- d. StringBuffer

Question 19

Not yet answered

Marked out of 1.00

Which Java collection guarantees traversal in insertion order?

- a. HashSet
- b. TreeSet
- c. LinkedHashSet
- d. ConcurrentSkipListSet

Question 20

Not yet answered

Marked out of 1.00

Which statement is TRUE about Java streams?

- a. Intermediate operations are lazy
- b. Stream operations always run sequentially
- c. Streams can be reused
- d. Terminal operations produce another stream