Midterm 3 Study Guide

Due No due date **Points** 25 **Questions** 25 **Time Limit** 30 Minutes **Allowed Attempts** Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 26	16 minutes	25 out of 25
LATEST	Attempt 27	20 minutes	20 out of 25
	Attempt 26	16 minutes	25 out of 25
	Attempt 25	18 minutes	24 out of 25
	Attempt 24	15 minutes	25 out of 25
	Attempt 23	21 minutes	24 out of 25
	Attempt 22	24 minutes	24 out of 25
	Attempt 21	18 minutes	24 out of 25
	Attempt 20	30 minutes	19 out of 25
	Attempt 19	20 minutes	23 out of 25
	Attempt 18	30 minutes	21 out of 25
	Attempt 17	30 minutes	23.5 out of 25
	Attempt 16	30 minutes	24 out of 25
	Attempt 15	30 minutes	21 out of 25
	Attempt 14	30 minutes	23 out of 25
	Attempt 13	24 minutes	22 out of 25
	Attempt 12	16 minutes	24 out of 25
	Attempt 11	17 minutes	19 out of 25
	Attempt 10	17 minutes	22 out of 25
	Attempt 9	20 minutes	20 out of 25
	Attempt 8	21 minutes	20 out of 25
	Attempt 7	25 minutes	21.5 out of 25
	Attempt 6	25 minutes	21 out of 25
	Attempt 5	30 minutes	17 out of 25
	Attempt 4	21 minutes	23 out of 25
	Attempt 3	26 minutes	19.89 out of 25
	Attempt 2	30 minutes	22.5 out of 25
	Attempt 1	27 minutes	21 out of 25

ⓘ Correct answers are hidden.

Submitted Jul 20 at 1:55pm

Question 1

1 / 1 pts

A loop that reads data until the input stream signals that it is done is called a sentinel loop.

- ○ True
- ◉ False

Question 2

1 / 1 pts

In the ***primed loop pattern***, you read data before the loop and at the end of the loop.

- ◉ True
- ○ False

Question 3

1 / 1 pts

A function template may be defined in a header file.

- ◉ True
- ○ False

Incorrect

Question 4

0 / 1 pts

What is true about this code?

```
template <typename T, typename U>
T pickle(T& a, const U& b) {
    a += b;
    return b;
}

int main()
{
    auto x = 42.5;
    auto y = pickle(x, 5);
    cout << x << endl;
    cout << y << endl;
}
```

- [ ] In main, x prints 47
- [x] In main, x prints 47.5
- [ ] In main, y prints 5
- [ ] No answer text provided.
- [x] In main, y prints 47.5

Question 5

1 / 1 pts

The `#if` preprocessor directive may compare `double` literals but not variables.

- ( ) True
- (•) False

Question 6

1 / 1 pts

The directives `#if defined(symbol)` and `#ifndef symbol` mean, essentially, the same thing.

- ( ) True
- (•) False

Question 7

1 / 1 pts

What happens when this code fragment runs?

```
istringstream in("12");
int n;
in >> n;
```

- ( ) It compiles, but fails to link
- ( ) It sets an error state in `in`.
- ( ) It does not compile.
- (•) n is set to 12
- ( ) It throws a runtime exception
- ( ) None of these

Question 8

1 / 1 pts

The standard library version of `sqrt(-2)` throws a runtime exception because there is no possible answer.

- ( ) True
- (•) False

Question 9

1 / 1 pts

In the ***flag-controlled-pattern***, you use Boolean variable to signal when the sentinel is found.

- (•) True
- ( ) False

Question 10

1 / 1 pts

`vector` subscripts begin at `0` and go up to the `vector size`.

- ○ True
- ◉ False

Question 11

1 / 1 pts

An unnamed (anonymous) function is called a(n):

- ○ functor
- ○ stub
- ◉ lambda
- ○ iterator
- ○ None of these

Incorrect

Question 12

0 / 1 pts

The C++ specific term for a collection of variables that have distinct names and types is a ***structure***.

- ○ True
- ◉ False

Question 13

1 / 1 pts

Assume that v contains `[1, 2, 3]`. The result of writing `cout << v.at(4);` throws a runtime exception.

- ◉ True
- ○ False

Incorrect

Question 14

0 / 1 pts

What does this code do?

```
int x = 0;
vector<int> v{1, 3, 2};
for (auto e : v) x += e;
cout << x << endl;
```

- ☐ Prints 6
- ☑ Finds the last element in v
- ☑ Prints 2
- ☐ Sums the elements in v
- ☐ Prints 0
- ☐ Finds the largest element in v

Incorrect

Question 15

0 / 1 pts

Assume `vector<int> v`; Writing `cout << v.front();` throws a runtime exception.

- ◉ True
- ○ False

Question 16

1 / 1 pts

A `vector` consists of indexed elements.

- ◉ True
- ○ False

Question 17 1 / 1 pts

The statement `v.insert(v.end(), 3)` is undefined because `end()` points past the last element in the `vector`.

- ○ True
- ◉ False

Question 18 1 / 1 pts

Which area of memory are local variables stored in?

- ○ Uninitialized Data
- ○ Text
- ◉ Stack
- ○ Heap
- ○ Initialized Data

Question 19 1 / 1 pts

Explicitly initializing an array like this: ***`int a[3] = {1, 2, 3};`*** requires the size to be the same or smaller than the number of elements supplied.

- ○ True
- ◉ False

Question 20 1 / 1 pts

Which of these lines correctly prints 3?

```
struct S {
    int a = 3;
    double b = 2.5;
};

S obj, *p = &obj;
```

- ○ `cout << *p.a << endl;`
- ○ `cout << p.a << endl;`
- ◉ `cout << (*p).a << endl;`
- ○ `cout << *(p).a << endl;`
- ○ `cout << *(p.a) << endl;`

Question 21 1 / 1 pts

Which array definition is initialized to all zeros?

```
int SIZE = 3;
int a1[SIZE];
int a2[3];
int a3[3]{};
int a4[] = {1, 2, 3};
int a5[3] = {1, 2};
```

- ◉ `a3`
- ○ `a1`
- ○ `a2`
- ○ `a5`
- ○ None of these

Question 22 1 / 1 pts

What is printed when you run this code?

```
int n{};
int *p;
*p = n;
cout << *p << endl;
```

- The address value where ***n*** is stored
- Will not compile
- No compilation errors, but undefined behavior when run
- The value 0 (stored in n)
- None of these

Incorrect

Question 23

0 / 1 pts

What is printed when you run this code?

```
int *n{nullptr};
cout << &n << endl;
```

- No compilation errors, but undefined behavior
- No output; compiler error.
- The address value where ***n*** is stored
- The word "`nullptr`"
- The address value `0`

Question 24

1 / 1 pts

What is the equivalent ***array notation***?

```
int dates[10];
cout << (*dates + 2) + 2 << endl;
```

- `dates[2]`
- `dates[0] + 4`
- `dates[0] + 2`
- `&dates[2]`
- `dates[2] + 2`

Question 25

1 / 1 pts

In C++ printing an array name prints the value of the first element in the array.

- True
- False