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 30	23 minutes	25 out of 25
LATEST	Attempt 32	24 minutes	23 out of 25
	Attempt 31	29 minutes	24 out of 25
	Attempt 30	23 minutes	25 out of 25
	Attempt 29	23 minutes	24 out of 25
	Attempt 28	22 minutes	25 out of 25
	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 5:54pm

Incorrect **Question 1** 0 / 1 pts

What is true about this piece of code?

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

int main()
{
    int x = 42;
    auto a = pickle(x, 4.5);
    cout << a << endl;
    cout << x << endl;
}
```

- [x] In main, x prints 46
- [ ] This code has a syntax error.
- [ ] In main, a prints 4
- [x] In main, a prints 4.5
- [ ] In main, x prints 46.5

Question 2 1 / 1 pts

You can report a logical error encountered in your code by using the `throw` keyword.

- (●) True
- ( ) False

Question 3

1 / 1 pts

If no exception is thrown in a `try` block, all `catch` blocks associated with that `try` block are ignored.

- (●) True
- ( ) False

Question 4

1 / 1 pts

In a sequence of `try/catch` blocks, the last `catch` block of that sequence should be ____.

- ( ) `catch(exception){}`
- ( ) `catch(int x){ }`
- (●) `catch(...){ }`
- ( ) `catch(str){ }`

Question 5

1 / 1 pts

Calling a template function like `to_string(3.5)` is known as implicit instantiation.

- (●) True
- ( ) False

Question 6

1 / 1 pts

Which of the following loop patterns are used here?

```
string s{"hello CS 150"};
for (auto e : s)
{
    if (toupper(e))
        out.put('x');
}
```

- (●) iterator or range loop
- ( ) limit loop
- ( ) loop-and-a-half
- ( ) inline test
- ( ) data loop
- ( ) sentinel loop
- ( ) counter-controlled loop
- ( ) primed loop

Question 7

1 / 1 pts

A function template may be declared in a header file but **must be** defined in an implementation file.

- ( ) True
- (●) False

Question 8

1 / 1 pts

Which of the following blocks is designed to catch any type of exception?

- ( ) `catch(exception){ }`

- `catch(){ }`
- `catch(...){ }`
- `catch(*){ }`

Question 9 1 / 1 pts

A completion code is a special return value that means "the function failed to execute correctly."

- True
- False

Question 10 1 / 1 pts

The declaration: `vector<string> v(5);` creates a `vector` containing five empty `string` objects.

- True
- False

Question 11 1 / 1 pts

Assume the `vector` `v` contains `[1, 2, 3]`. `v.erase(0);` is a syntax error.

- True
- False

Incorrect

Question 12 0 / 1 pts

The C++ specific term for classes like `vector` are generic classes.

- True
- False

Question 13 1 / 1 pts

Each element in a `vector` may be of a different type.

- True
- False

Question 14 1 / 1 pts

In the declaration: `vector<int> v;` the word `vector` represents the object's ***base type***.

- True
- False

Question 15 1 / 1 pts

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

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

Question 16 1 / 1 pts

Match each item with the correct statement below.

Removes the first element in v and shifts the rest to the left	v.erase(v.begin())
Returns a reference to the last element in v	v.back()
Creates the vector [2, 3]	vector<int> v{2, 3};
Points to the first element in v	v.begin()

Question 17

1 / 1 pts

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

- ○ True
- ◉ False

Question 18

1 / 1 pts

All of these are legal C++ statements; which of them uses the C++ ***dereferencing operator***?

```
int a = 3, b = 4;
```

- ○ None of these use the dereferencing operator.
- ◉ `int x = *p;`
- ○ `z *= a;`
- ○ `int y = a * b;`
- ○ `int *p = &b;`

Question 19

1 / 1 pts

The variable ***buf*** is a pointer to a region of memory storing contiguous ***int*** values. (This is similar to your homework, where you had a region of memory storing ***unsigned char*** values.) The four lines shown here are legal. ***Which operation is legal***?

```
int *p1 = buf;
const int *p2 = buf;
int * const p3 = buf;
const int * p4 const = buf;
```

- ◉ `*p3 = 5;`
- ○ `p3++;`
- ○ `*p4 = 7;`
- ○ `p4++;`
- ○ `*p2 = 3;`

Question 20

1 / 1 pts

In C++ assigning one array to another is permitted.

- ○ True
- ◉ False

Question 21

1 / 1 pts

Which expression returns the number of countries?

```
string countries[] = {"Andorra", "Albania", . . . };
```

- ◉ `sizeof(countries) / sizeof(countries[0])`
- ○ `len(countries)`
- ○ None of these
- ○ `sizeof(countries) * sizeof(countries[0])`

- `sizeof(countries)`

Question 22

1 / 1 pts

Examine the following code. What is stored in ***a*** after it runs.

```
int f(int * p, int x)
{
    *p = x * 2;
    return x / 2;
}
. . .
int a = 3, b, c;
c = f(&b, a);
```

- 1
- 2
- 3
- 6
- Does not compile

Question 23

1 / 1 pts

Here is the pseudocode for the ***greenScreen()*** function from your homework. What single statement sets the red, green and blue components to **0**?

```
Let p point the beginning of the image
Set end to point just past the end
While p != end
    If *(p + 3) is 0 (transparent)
        Clear all of the fields
    Increment p by 4
```

- `&(p + 1) = &(p + 2) = &(p + 3) = 0;`
- `p = p + 1 = p + 2 = 0;`
- None of these
- `*(p) = *(p + 1) = *(p + 2) = 0;`
- `*(p + 1) = *(p + 2) = *(p + 3) = 0;`

Question 24

1 / 1 pts

What is the term used to describe a variable which stores a memory address?

- `lvalue`
- None of these
- `pointer`
- `rvalue`
- `reference`

Question 25

1 / 1 pts

Examine this version of the ***swap()*** function. How do you call it?

```
void swap(int& x, int * y)
{
    . . .
}
. . .
int a = 3, b = 7;
// What goes here ?
```

- None of these
- `swap(a, &b);`
- `swap(&a, b);`
- `swap(&a, &b);`
- `swap(a, b);`