


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 4	21 minutes	23 out of 25
LATEST	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 13 at 5:50am

Question 11 / 1 pts

Which fragment completes this code segment?

```
string fmt(double n, int decimals)
{
    ostringstream out;
    out << fixed << setprecision(decimals);
    out << n;
    return _____;
}
```

- ☐ out
- ☐ n
- ☒ out.str()
- ☐ None of these
- ☐ out.to_string()

IncorrectQuestion 20 / 1 pts

What prints?

```
string s("hello");
try {
    if (s.size() > 20) throw 42;
    if (islower(s.back())) throw "goodbye";
    if (s == "hello") throw string("hello");
    s.at(s.size()) = 'x';
    cout << "one\n";
}
catch (const int& e) { cout << "two\n"; }
catch (const string& e) { cout << "three\n"; }
catch (exception& e) { cout << "four\n"; }
catch (...) { cout << "five\n"; }
```

- ☐ four
- ☐ five
- ☐ one
- ☐ two
- ☒ three
- ☐ Undefined

IncorrectQuestion 30 / 1 pts

Complete the code fragment below, which is designed to throw an illegal_length exception if string variable accountNumber has more than seven characters.

```
if (accountNumber.size() > 7)
{
    _____;
}
```

}

☒ throws illegal_length("Account number exceeds maximum length");

☐ throws new illegal_length("Account number exceeds maximum length");

☐ throw illegal_length("Account number exceeds maximum length");

☐ throw new illegal_length("Account number exceeds maximum length");

Question 4

1 / 1 pts

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

☐ True

☒ False

Question 5

1 / 1 pts

When you throw an exception, control immediately jumps out of the current try block.

☒ True

☐ False

Question 6

1 / 1 pts

Which of these are appropriate uses of the C++ `assert` facility?

☒ Debugging checks

☒ Validate assumptions about your code

☒ Validate function arguments under the programmer's control

☐ Validate input received by your program

☒ Validate the postcondition of a calculation

☐ Error conditions (such as file not found)

Question 7

1 / 1 pts

A specialized error handling block of code, is called a try block.

☐ True

☒ False

Question 8

1 / 1 pts

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

☐ True

☒ False

Question 9

1 / 1 pts

A `catch` block specifies the type of exception it can catch and immediately terminates the program.

☐ True

☒ False

Incorrect

Question 10

0 / 1 pts

What prints when this code runs?

```
enum class Coin
{
    PENNY = 1, NICKEL = 5, DIME = 10, QUARTER = 25
};
```

```
Coin c = NICKEL;
cout << static_cast<int>(c) << endl;
```

- ☒ 5
- ☐ Does not compile; Missing semicolon at end of list of members.
- ☐ Does not compile; Cannot assign NICKEL to c.
- ☐ 2

Question 11

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 12

1 / 1 pts

Which statement is false? The elements in a vector:

- ☐ are homogeneous
- ☒ None of these
- ☐ are are all of the same type
- ☐ each use the same amount of memory
- ☐ are accessed by using an index or subscript

Question 13

1 / 1 pts

Which line prints 3?

```
int main()
{
    vector<int> v{1, 2, 3};
    auto size = v.size();

    cout << v.back() << endl;      // 1.
    cout << v.front() << endl;     // 2.
    cout << v.at(0) << endl;       // 3.
    cout << v.at(size) << endl;    // 4.
    cout << v.pop_back() << endl;  // 5.
}
```

- ☐ 2
- ☐ 5
- ☐ 4
- ☐ 3
- ☒ 1

Question 14

1 / 1 pts

The pop_back member function adds elements to the end of a vector.

- ☐ True
- ☒ False

Question 15

1 / 1 pts

This is the correct syntax for a C++ scoped enumeration.

```
enum class WEEKEND {SUNDAY, SATURDAY=6};
```

☒ True

☐ False

Question 161 / 1 pts

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

☐ iterator

☐ stub

☐ None of these

☒ lambda

☐ functor



Incorrect

Question 170 / 1 pts

A vector consists of named members.

☒ True

☐ False

Question 181 / 1 pts

The value for the variable *b* is stored:

```
int a = 1;
void f(int b)
{
    int c = 3;
    static int d = 4;
}
```

☐ on the heap

☐ in the static storage area

☐ The example does not provide enough information

☐ in the CPU machine registers

☒ on the stack

Question 191 / 1 pts

An incomplete type and a forward reference generally mean the same thing.

☒ True

☐ False

Incorrect

Question 200 / 1 pts

What is the equivalent *array notation*?

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

☐ dates[0] + 2

☒ dates[0] + 4

☐ dates[2] + 2

☐ dates[2]

☐ &dates[2]

Question 211 / 1 pts

The elements of a C++ *string* array with no explicit initialization, created in a function will be set to *null*.

☐ True

☒ False

Question 221 / 1 pts

Which of these lines correctly prints 2.5?

```
struct S {  
    int a = 3;  
    double b = 2.5;  
};  
  
S obj, *p = &obj;
```

☐ cout << *(p).b << endl;

☒ cout << p->b << endl;

☐ cout << *p.b << endl;

☐ cout << *p->b << endl;

☐ cout << *(p.b) << endl;



Question 231 / 1 pts

You must use an integral constant or literal to specify the size of a built-in C++ array.

☒ True

☐ False

Incorrect

Question 240 / 1 pts

In C++ using == to compare one array to another is illegal.

☒ True

☐ False

Question 251 / 1 pts

What is true about this code?

```
int n{500};  
int *p = &n;
```

☒ *p is the value of n

☐ p stores the same value as n

☐ &p is the direct or explicit value of n

☐ &p represents the indirect value of n

☐ &n is the indirect value of p