

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 24	15 minutes	25 out of 25
LATEST	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 1:02pm

Question 1

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 2

1 / 1 pts

Match each item with the correct loop form below.

Indefinite limit loop that reduces its input	<code>while (n != 0) { n /= 2; }</code> ▾
Indefinite limit loop that uses successive approximations	<code>while(abs(g1 – g2) >= EPSILON)</code> ▾
Counter-controlled symmetric loop for producing a sequence of data	<code>for (int i=12; i <= 19; i++) { . . . }</code> ▾
Indefinite data loop that uses raw input	<code>while(cin.get(ch)) { . . . }</code> ▾
Counter-controlled asymmetric loop for processing characters	<code>for (size_t i=0, len=s.size(); i < l</code> ▾
Iterator loop that may change its container	<code>for (auto& e : col) { . . . }</code> ▾
Iterator loop that cannot change its container	<code>for (auto e : col) { . . . }</code> ▾
Counter-controlled loop for processing substrings	<code>for (size_t i=4, slen=4, len=s.siz</code> ▾
Indefinite data loop that uses formatted input	<code>while(cin >> n) { . . . }</code> ▾

Question 3

1 / 1 pts

The statement `#if abs(-3) > 2` is legal.

☐ True

☒ False

Question 41 / 1 pts

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

☐ True

☒ False

Question 51 / 1 pts

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

☐ True

☒ False

Question 61 / 1 pts

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

☒ True

☐ False

Question 71 / 1 pts

What happens when this code fragment runs?

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

☐ It compiles, but fails to link

☐ It does not compile.

☐ It throws a runtime exception

☐ It sets an error state in `in`.

☐ None of these

☒ `n` is set to 12

Question 81 / 1 pts

A function template may be **defined** in a header file.

☒ True

☐ False

Question 91 / 1 pts

The heading of a `try` block can contain ellipses in place of a parameter.

☐ True

☒ False

Question 101 / 1 pts



Assume `vector<int> v`; Writing `cout << v.front()`; is undefined.

☒ True

☐ False

Question 111 / 1 pts

A vector consists of named members.

☐ True

☒ False

Question 121 / 1 pts

The following is legal. Which is the correct way to access a data member in the Rectangle variable named r?

```
struct Rectangle { int length, width; };
```

☐ r{length}

☐ r->length

☒ r.length

☐ Either r.length or r->length will work

☐ r[0]

☐ None of these are correct

Question 131 / 1 pts

What is stored in data after this runs?

```
vector<int> data{1, 2, 3};  
data.pop_back();
```

☐ [2, 3]

☐ [1, 2, 3]

☐ [1, 2, 3, 0]

☐ None of these

☒ [1, 2]

☐ []



Incorrect

Question 140 / 1 pts

Which statement is false? The elements in a vector:

☐ are stored next to each other in memory

☐ are all of the same type

☐ are homogeneous

☒ None of these

☐ are accessed by name

Question 151 / 1 pts

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

☒ True

☐ False

Question 161 / 1 pts

Assume the vector v contains [1, 2, 3]. v.erase(v.begin()); changes v to [2, 3].

☒ True

☐ False

Question 171 / 1 pts

The statement `v.insert(v.begin(), 3)` inserts the element 3 into the vector `v`, overwriting the exiting element at index 0.

☐ True

☒ False

Question 181 / 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 ?
```

☐ `swap(&a, &b);`

☐ None of these

☒ `swap(&a, b);`

☐ `swap(a, &b);`

☐ `swap(a, b);`

Question 191 / 1 pts

In C++ initializing an array with the contents of another is permitted.

☐ True

☒ False

Question 201 / 1 pts

Which array definition is illegal (even if it may compile on some compilers)?

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

☒ `a1`

☐ None of these

☐ `a3`

☐ `a2`

☐ `a5`

Question 211 / 1 pts

What is the equivalent *array notation*?

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

☐ `dates[0] + 4`

☐ `dates[2]`

☒ `dates[0] + 2`

☐ `&dates[2]`

☐ `dates[2] + 2`



Question 22

1 / 1 pts

The elements of a C++ array created outside of a function are allocated on the stack.

☐ True

☒ False

Question 23

1 / 1 pts

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

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

☐ Does not compile

☒ 6

☐ 3

☐ 2

☐ 1

Question 24

1 / 1 pts

The value for the variable *a* is stored:

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

☒ in the static storage area

☐ on the heap

☐ on the stack

☐ in the CPU machine registers

☐ The example does not provide enough information

Question 25

1 / 1 pts

Which returns the last pixel on the first row of this image?

```
Pixel *p;    // address of pixel data
int w, h;    // width and height of image
```

☐ None of these are correct

☐ *p[w - 1]

☒ p[w - 1]

☐ p[w] - 1

☐ p + w - 1

