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 12	16 minutes	24 out of 25
LATEST	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 13 at 7:10am

Question 1

1 / 1 pts

To use strings as a data stream source or sink, use the `<sstream>` header

- (●) True
- ( ) False

Incorrect

Question 2

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

Question 3

1 / 1 pts

The `try` block is followed by one or more ____ blocks.

- ( ) `throw`
- ( ) `do`
- (●) `catch`
- ( ) `finally`

Question 4 1 / 1 pts

A `catch` block is a block of code where runtime or logical errors may occur.

- ○ True
- ◉ False

Question 5 1 / 1 pts

The `class` ____ is designed to deal with illegal arguments used in a function call.

- ○ `bad_argument`
- ○ `invalid_call`
- ○ `illegal_argument`
- ◉ `invalid_argument`

Question 6 1 / 1 pts

The `logic_error` and `runtime_error` classes are defined in the header file ____.

- ◉ `stdexcept`
- ○ `exception`
- ○ `stdex`
- ○ `stdlib`

Incorrect **Question 7** 0 / 1 pts

What happens with the following section of code?

```
if (__APPLE__)
   cout << "Running on a Mac" << endl;
else if (__WIN32)
   cout << "Running on Windows" << endl;
else if (__linux)
   cout << "Running on Linux" << endl;
else
   cout << "Running on an unknown platform" << endl;
```

- ○ The program will not compile
- ◉ All lines will be included in the program. It will print the platform you are running on.
- ○ Only the lines that identify your platform will be included in the executable
- ○ The program will crash if compiled on one platform, but run on another.

Question 8 1 / 1 pts

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

- ○ True
- ◉ False

Question 9 1 / 1 pts

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

- ◉ True
- ○ False

Question 10 1 / 1 pts

The built-in primitive data types such as `int`, char and `double` are ***structured*** data types.

- ○ True
- ◉ False

Question 11

1 / 1 pts

You may create a structure variable as part of a structure definition.

- (●) True
- ( ) False

Question 12

1 / 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;
```

- [x] Sums the elements in v
- [ ] Prints 2
- [ ] Finds the last element in v
- [x] Prints 6
- [ ] Prints 0
- [ ] Finds the largest element in v

Question 13

1 / 1 pts

Assuming that you have an iterator named `iter`. Which statement retrieves the element that `iter` refers to?

- ( ) `auto d = iter;`
- ( ) `auto c = &iter;`
- (●) `auto b = *iter;`
- ( ) None of these
- ( ) `auto a = ++iter;`

Question 14

1 / 1 pts

What prints?

```
vector<int> v{1, 2, 3, 4, 5};
v.pop_back();
cout << v.front() << endl;
```

- ( ) 5
- ( ) Nothing; run-time error.
- ( ) Nothing; compile-time error.
- (●) 1
- ( ) 4

Question 15

1 / 1 pts

Examine the following code (which is legal). What changes are necessary to allow the statement `if (m1 != m2)` … to compile?

```
struct Money { int dollars{0}, cents{0}; } m1, m2;

bool equals(const Money& lhs, const Money& rhs)
{
    return lhs.cents == rhs.cents &&
        lhs.dollars == rhs.dollars;
}
```

- ( ) The name of `equals()` must be changed to `operator==`
- ( ) This is not possible in C++.
- ( ) The function `equals()` must be named `notEquals()`.
- ( ) The type `Money` needs to be a `class`

- You must write a function named `operator!=`

Question 16

1 / 1 pts

The following code is ***legal***.

```
struct {int hours, seconds; } MIDNIGHT{0, 0};
```

- True
- False

Question 17

1 / 1 pts

Examine the following code (which is legal). Which statement is correct?

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

- `Rectangle r;`
- `Rectangle r = new Rectangle();`
- None of these are correct
- `r Rectangle;`
- `Rectangle r();`

Question 18

1 / 1 pts

Which line creates an array with 5 elements?

- `int a[4];`
- `int[5] d;`
- `int[] c[5];`
- `int b[5];`
- None of these

Question 19

1 / 1 pts

Assume that ***p1*** is a pointer to an integer and ***p2*** is a pointer to a second integer. Both integers appear inside a large contiguous sequence in memory, with ***p2*** storing a larger address. How many total integers are there in the slice between ***p1*** and ***p2***?

- `p2 - p1;`
- `p1 - p2;`
- `p1 - p2 + 1;`
- `p2 - p1 - 1;`
- None of these

Question 20

1 / 1 pts

Assume that ***ppi*** correctly points to ***pi***. Which line prints the value stored inside ***pi***?

```
int main()
{
    double pi = 3.14159;
    double *ppi;
    // code goes here
    // code goes here
}
```

- `cout << *pi;`
- `cout << &pi;`
- None of these
- `cout << ppi;`
- `cout << &ppi;`

Question 21

1 / 1 pts

What is printed when you run this code?

```
int n{};
int *p = &n;
*p = 10;
n = 20;
cout << *p << endl;
```

- 10
- The address of n
- 0
- **20**
- None of these

Question 22

1 / 1 pts

In C++ using == to compare one array to another is permitted (if meaningless).

- **True**
- False

Question 23

1 / 1 pts

Which assigns a value to the first position in ***letters***?

```
char letters[26];
```

- letters = 'a';
- letters[1] = 'b';
- letters[0] = "a";
- **letters[0] = 'a';**
- letters.front() = 'a';

Question 24

1 / 1 pts

C++ arrays can be allocated with a size of ***0***.

- True
- **False**

Question 25

1 / 1 pts

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

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

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