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 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 12 at 3:41pm

Question 1

1 / 1 pts

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

- (•) True
- ( ) False

Question 2

1 / 1 pts

Programmers use _____ to reason about logical correctness in their code.

- ( ) completion codes
- (•) assertions
- ( ) exceptions
- ( ) assumptions

Question 3

1 / 1 pts

Match each item with the correct standard header below.

Read and write characters to memory using streams	sstream
Connect a disk file to an input or output stream.	fstream
Use the predefined stream objects `cin` and `cout`	iostream
Determine the category of a character	cctype
Modify the way that memory is converted to characters on input or output	iomanip

Question 4

1 / 1 pts

What happens when this code fragment runs?

```
istringstream in(".5");
int n;
in >> n;
```

- (•) It sets an error state in `in`.
- ( ) It compiles, but fails to link
- ( ) It throws a runtime exception
- ( ) None of these
- ( ) It does not compile.
- ( ) `n` is set to `5`

Incorrect

Question 5

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;
}
```

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

Question 6

1 / 1 pts

To deal with errors in a program, such as a string subscript out of range or an invalid argument to a function call, several classes are derived from the `class` ____.

- ( ) `exception`
- ( ) `logic_exception`
- ( ) `runtime_error`
- (•) `logic_error`

Question 7

1 / 1 pts

Match each item with the correct standard header below.

Read and write characters to memory using streams	sstream
Connect a disk file to an input or output stream.	fstream
Use the predefined stream objects `cin` and `cout`	iostream
Determine the category of a character	cctype
Modify the way that memory is converted to characters on input or output	iomanip

Question 8

1 / 1 pts

The `class` ____ is the base of the classes designed to handle exceptions.

- ( ) `class`
- (•) `exception`
- ( ) `logic_error`
- ( ) `runtime_error`

Question 9

1 / 1 pts

Functions with generic (or type) parameters are known as template functions.

- (•) True
- ( ) False

Question 10

1 / 1 pts

The declaration: `vector<string> v{"bill", "bob", "sally"};` creates a `vector` containing three `string` objects.

- (●) True
- ( ) False

Question 11

1 / 1 pts

The structure and variable definitions are fine. Which statements are legal?

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

- (●) `cin >> little.width;`
- ( ) `cout << Rectangle.length;`
- ( ) None of these are correct
- ( ) `cin >> big;`
- ( ) `double p = 2 * (length + width);`

Question 12

1 / 1 pts

What is stored in `data` after this runs?

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

- ( ) `[1, 2, 3, 0]`
- ( ) None of these
- (●) `[1, 2]`
- ( ) `[]`
- ( ) `[2, 3]`
- ( ) `[1, 2, 3]`

Question 13

1 / 1 pts

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

- (●) True
- ( ) False

Question 14

1 / 1 pts

The declaration: `vector<int> v(10);` creates a `vector` object containing uninitialized elements.

- ( ) True
- (●) False

Question 15

1 / 1 pts

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

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

- ( ) `m2.cents++;`
- ( ) `if (m1.cents != m2.dollars) . . .`
- (●) `cout << m1 << endl;`
- ( ) `m1 = m2;`

Incorrect

Question 16

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

Question 17

1 / 1 pts

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

- ◉ True
- ○ False

Question 18

1 / 1 pts

What is true about an uninitialized pointer?

- ○ It is set to the `nullptr` value
- ○ Dereferencing it will cause a program crash
- ○ Dereferencing it is safe, but has no effect.
- ○ None of these are true
- ◉ Dereferencing it is undefined behavior

Question 19

1 / 1 pts

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

- ◉ True
- ○ False

Question 20

1 / 1 pts

What is true about this code?

```
int * choice;
```

- ○ `choice` can point to any kind of object
- ◉ `choice` contains an undefined address
- ○ Syntax error; should be `int choice*;`
- ○ `choice` currently contains an integer
- ○ `choice` currently points to an integer

Question 21

1 / 1 pts

The value for the variable ***b*** is stored:

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

- ○ in the CPU machine registers
- ○ in the static storage area
- ◉ on the stack

- on the heap
- The example does not provide enough information

Question 22

1 / 1 pts

You can calculate the number of elements in an array, provided the array definition is in scope.

- (•) True
- ( ) False

Question 23

1 / 1 pts

The value for the variable ***c*** is stored:

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

- ( ) in the CPU machine registers
- ( ) in the static storage area
- ( ) on the heap
- (•) on the stack
- ( ) The example does not provide enough information

Question 24

1 / 1 pts

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

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

- ( ) `dates[2] + 2`
- ( ) `dates[0] + 4`
- ( ) `&dates[2]`
- ( ) `dates[0] + 2`
- (•) `dates[2]`

Question 25

1 / 1 pts

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

- (•) True
- ( ) False