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 22	24 minutes	24 out of 25
LATEST	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 19 at 1:17pm

Question 1

1 / 1 pts

Match each item with the correct loop form below.

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

Question 2

1 / 1 pts

In the ***loop-and-a-half***, you use a `break` statement to exit the loop when the sentinel is found.

- (•) True
- ( ) False

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 you execute the following (erroneous) code:

```
cout << stoi(42.5) << endl;
```

- (●) The code does not compile because the argument is the wrong type.
- ( ) No conversion takes place and the output stream is placed in a failed state.
- ( ) The program prints an error message and terminates since you cannot convert a double to an int
- ( ) The double 42.5 is truncated to 42 and printed
- ( ) An exception is thrown, which may be caught

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

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 = 42.0;
    auto y = pickle(x, 4.5);
    cout << x << endl;
    cout << y << endl;
}
```

- [x] In main, x prints 46.5
- [ ] This code does not compile.
- [ ] In main, x prints 46
- [x] In main, y prints 4.5
- [ ] In main, y prints 4

Question 7

1 / 1 pts

What happens when this code fragment runs?

```
istringstream in("12.5");
int n;
in >> n;
```

- ( ) It sets an error state in `in`.
- ( ) None of these

- It compiles, but fails to link
- n is set to 12
- It throws a runtime exception
- It does not compile.

Question 8

1 / 1 pts

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

- True
- False

Question 9

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 10

1 / 1 pts

User-defined scalar types are created with the `enum class` keywords in C++.

- True
- False

Question 11

1 / 1 pts

An _____ is an object which specifies the position of an element inside a container, regardless of what kind of container you use.

- iterator
- index
- subscript
- pointer
- lambda

Incorrect

Question 12

0 / 1 pts

Assume that v contains `[1, 2, 3]`. The result of writing `cout << v[4];` throws a runtime exception.

- True
- False

Question 13

1 / 1 pts

`vector` subscripts begin at `0` and go up to the `vector size - 1`.

- True
- False

Question 14

1 / 1 pts

What prints when this code runs?

```
enum class Coin
{
   PENNY = 1, NICKEL, DIME, QUARTER
};
cout << static_cast<int>(Coin::DIME) << endl;
```

- 2
- 10
- Does not compile; Missing semicolon at end of list of members.
- 3

Question 15

1 / 1 pts

In C++, objects have ***value semantics***; structure variables contain the data members.

- True
- False

Question 16

1 / 1 pts

Examine the following code. Which element is erased?

```
vector<int> v{1, 2, 3};
v.erase(begin(v), end(v));
```

- Does not compile
- 2
- 3
- 1
- All the elements are erased

Question 17

1 / 1 pts

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

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

- `if (m1.dollars > m2.cents) …`
- `cout << m1 << endl;`
- `if (m1 != m2) . . .`
- `m1 = {3, 4};`

Question 18

1 / 1 pts

What is a common pointer error?

- Using a pointer without first initializing it
- Assigning a new value to a pointer
- Setting a pointer value to `nullptr`
- Dereferencing a pointer
- Using indirection on a pointer

Question 19

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;
}
```

- The example does not provide enough information

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

Question 20

1 / 1 pts

What does the array ***a*** contain after this runs?

```
int a[] = {1, 2, 3};
int b[] = {4, 5, 6};
a = b;
```

- ○ Undefined behavior
- ○ `{4, 5, 6}`
- ○ `{1, 2, 3}`
- ◉ Syntax error; does not compile.

Question 21

1 / 1 pts

What is printed?

```
int a[] = {1, 2, 3};
int b[] = {1, 2, 3};

if (a == b) cout << "a == b" << endl;
else cout << "a != b" << endl;
```

- ◉ `a != b`
- ○ `a == b`
- ○ Syntax error; does not compile.
- ○ Undefined behavior

Question 22

1 / 1 pts

Which array definition is initialized to all zeros?

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

- ○ `a5`
- ○ `a2`
- ○ None of these
- ◉ `a3`
- ○ `a1`

Question 23

1 / 1 pts

If ***`size_t len = 0`***; then ***`len - 1`*** is the largest possible unsigned number.

- ◉ True
- ○ False

Question 24

1 / 1 pts

In C++ an array variable and the array elements are separate. The array variable contains the address of the first element in the array.

- ○ True
- ◉ False

Question 25 1 / 1 pts

A forward reference can be used when you want to use a structure as a data member without first defining the entire structure.

- ○ True
- ◉ False