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 30	23 minutes	25 out of 25
LATEST	Attempt 31	29 minutes	24 out of 25
	Attempt 30	23 minutes	25 out of 25
	Attempt 29	23 minutes	24 out of 25
	Attempt 28	22 minutes	25 out of 25
	Attempt 27	20 minutes	20 out of 25
	Attempt 26	16 minutes	25 out of 25
	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 5:25pm

Question 1

1 / 1 pts

What happens when this code fragment runs?

```
cout << stoi("12") << endl;
```

○ It does not compile.

○ It compiles, but fails to link

○ None of these

● `stoi()` returns 12

○ It throws a runtime exception

○ It sets an error state in `cout`.

Question 2

1 / 1 pts

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

○ True

● False

Question 3

1 / 1 pts

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

● True

○ False

Question 4

1 / 1 pts

What happens when you execute the following (erroneous) code:

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

○ The program prints an error message and terminates since you cannot convert a double to an int

● 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 double 42.5 is truncated to 42 and printed

○ An exception is thrown, which may be caught

Question 5

1 / 1 pts

The line: `ifstream in("x");` throws a runtime exception if a file `x` cannot be found.

○ True

● False

Question 6

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 7

1 / 1 pts

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

● True

○ False

Question 8

1 / 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;
```

○ All lines will be included in the program. It will print the platform you are running on.

● The program will not compile

○ The program will crash if compiled on one platform, but run on another.

○ Only the lines that identify your platform will be included in the executable

Question 9

1 / 1 pts

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

● `exception`

○ `logic_error`

○ `class`

○ `runtime_error`

Question 10

1 / 1 pts

What prints?

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

○ Nothing; run-time error.

○ 5

○ 1

● 4

○ Nothing; compile-time error.

Question 11

1 / 1 pts

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

● True

○ False

Incorrect

Question 12

0 / 1 pts

What prints when this code runs?

```
enum class Coin
{
   PENNY, NICKEL, DIME, QUARTER
};
cout << Coin::PENNY << endl;
```

○ `Coin::PENNY`

○ `0`

● Does not compile; Cannot output enumerated members without overloaded operator.

○ `PENNEY`

○ `1`

Question 13

1 / 1 pts

Given the following structure and variable definitions, which data members are ***default initialized***?

```
struct Employee
{
   long empID;
   std::string lastName;
   double salary;
   int age;
};

Employee bob{777, "Zimmerman"};
```

☑ `salary`

□ `empID`

□ `lastName`

□ None of these

☑ `age`

Question 14

1 / 1 pts

Assuming that `Star` is a structure, the declaration: `vector<Star> stars(3);` creates three uninitialized `Star` objects.

○ True

● False

Question 15

1 / 1 pts

Assume that v contains `[1, 2, 3]`. The result of writing `cout << v.at(4);` is a compiler error.

○ True

● False

Question 16

1 / 1 pts

Structures are ***heterogeneous*** data types.

● True

○ False

Question 17

1 / 1 pts

Given the following structure and variable definitions, which data members are ***default initialized***?

```
struct Employee
{
   long empID;
   std::string lastName;
   double salary;
   int age;
};

Employee bob{777, "Zimmerman", 5000000.0, 76};
```

● None of these

○ `salary`

○ `lastName`

○ `age`

○ `empID`

Question 18

1 / 1 pts

Which of these lines displays the eighth element of ***a***?

```
int a[15];
```

○ `cout << a[8] << endl;`

○ `cout << a(7) << endl;`

○ `cout << a.at(7) << endl;`

● `cout << a[7] << endl;`

Question 19

1 / 1 pts

In C++ assigning one array to another is permitted.

○ True

● False

Question 20

1 / 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};
```

○ a5

○ a2

○ a3

● a1

○ None of these

Question 21

1 / 1 pts

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

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

● in the static storage area

○ on the stack

○ on the heap

○ in the CPU machine registers

○ The example does not provide enough information

Question 22

1 / 1 pts

Which array definition is illegal?

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

○ a3

○ a2

○ None of these

● a5

○ a1

Question 23

1 / 1 pts

Explicitly initializing an array like this: `int a[] = {1, 2, 3};` only works in C++ 11.

○ True

● False

Question 24

1 / 1 pts

What is printed when you run this code?

```
int *n{nullptr};
cout << n << endl;
```

○ The word "`nullptr`"

● The address value `0`

○ No output; compiler error.

○ No compilation errors, but undefined behavior

Question 25

1 / 1 pts

Array subscripts are not range checked

● True

○ False