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 2	30 minutes	22.5 out of 25
LATEST	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 3:07am

Question 1

1 / 1 pts

What happens when this code fragment runs in C++ 11?

```
cout << sqrt(-2) << endl;
```

- ○ It sets an error state in `cout`.
- ○ None of these
- ○ It throws a runtime exception
- ◉ `sqrt()` returns a not-a-number error value
- ○ It does not compile.
- ○ `-1.41421` is printed

Partial

Question 2

0.5 / 1 pts

Suppose you have the following code, and `x` is negative. What happens?

```
cout << sqrt(x) * sqrt(x) << endl;
```

- ☐ The program will terminate with an assertion, since you can't take the square root of a negative number.
- ☐ The global error variable, `errno`, may be set to indicate the error
- ☐ The program will throw an exception
- ☐ The program will not compile
- ☐ The original value of `x` is printed
- ☑ The completion code value, `nan`, may be printed signaling an error.

Question 3

1 / 1 pts

What happens with the following section of code?

```
cout << "Enter 1, 2 or 3: ";
int n;
cin >> n;
#if 1
    cout << "You entered 1" << endl;
#elif 2
    cout << "You entered 2" << endl;
#elif 3
    cout << "You entered 3" << endl;
#else
    cout << "Invalid value" << endl;
#endif
```

- ○ Does not compile
- ○ Compiles, but only prints "Invalid value"
- ○ Compiles and prints the correct value entered by the user.
- ◉ Compiles, but always print "You entered 1"

Question 4

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 5

1 / 1 pts

In the ***flag-controlled-pattern***, you use Boolean variable to signal when the sentinel is found.

- (●) True
- ( ) False

Question 6

1 / 1 pts

In a sequence of `try/catch` blocks, the last `catch` block of that sequence should be ____.

- (●) `catch(...){ }`
- ( ) `catch(exception){}`
- ( ) `catch(int x){ }`
- ( ) `catch(str){ }`

Question 7

1 / 1 pts

The C++11 standard library provides the function `stoi()` to convert a string to an integer. Which library is it found in?

- ( ) `cnvt`
- ( ) `iostream`
- ( ) None of these
- ( ) `cmath`
- (●) `string`

Question 8

1 / 1 pts

What happens when this code fragment compiles and runs?

```
#define N
#ifdef N
    cout << "Hello";
#else
    cout << "Goodbye";
#endif
```

- ( ) prints nothing
- (●) `prints "Hello"`
- ( ) `prints "HelloGoodbye"`
- ( ) `prints "Goodbye"`
- ( ) It does not compile.

Question 9

1 / 1 pts

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

- (•) `stdexcept`
- ( ) `stdex`
- ( ) `stdlib`
- ( ) `exception`

Question 10

1 / 1 pts

In C++ the parameterized collection classes are called _____?

- (•) templates
- ( ) collections
- ( ) enumerations
- ( ) abstract data types
- ( ) generics
- ( ) None of these

Incorrect

Question 11

0 / 1 pts

What prints when this code runs?

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

- ( ) `0`
- ( ) `PENNEY`
- ( ) `1`
- (•) Does not compile; Cannot output enumerated members without overloaded operator.
- ( ) `Coin::PENNY`

Question 12

1 / 1 pts

Which of these are true?

```
int main()
{
    vector<int> v{1, 2, 3};
    for (auto& e : v) e = 0;
    cout << v.at(0) << endl;
}
```

- [x] Prints `0`
- [ ] Code compiles but gives a warning
- [ ] Prints `1`
- [ ] Code runs but has no effect on `v`
- [ ] Prints `3`
- [ ] Code will not compile

Question 13

1 / 1 pts

Which defines a `vector` to store the salaries of ten employees?

- (•) `vector<double> salaries(10);`
- ( ) `vector<double> salaries{10};`
- ( ) `vector salaries(10);`
- ( ) `double salaries[10];`
- ( ) `vector<double> salaries[10];`
- ( ) None of these

Question 14

1 / 1 pts

What is stored in `data` after this runs?

```
vector<int> data{1, 2, 3};
data.push_back(0);
```

- ○ None of these
- ○ `[2, 3]`
- ○ `[1, 2]`
- ◉ `[1, 2, 3, 0]`
- ○ `[]`
- ○ `[1, 2, 3]`

Question 15

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;
```

- ☐ Sums the elements in v
- ☐ Prints 6
- ☑ Finds the last element in v
- ☐ Prints 0
- ☑ Prints 2
- ☐ Finds the largest element in v

Question 16

1 / 1 pts

Assume that `v` contains `[1, 2, 3]`. The result of writing `cout << v[4];` is undefined.

- ◉ True
- ○ False

Question 17

1 / 1 pts

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

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

- ○ `if (big.length == width) . . .`
- ○ `if (big != small) . . .`
- ○ None of these are correct
- ◉ `if (big.length == small.width) . . .`
- ○ `if (big == small) . . .`

Question 18

1 / 1 pts

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

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

- ○ `&dates[2]`
- ○ `dates[2] + 2`
- ○ `dates[0] + 4`
- ◉ `dates[2]`
- ○ `dates[0] + 2`

Question 19

1 / 1 pts

What is printed when you run this code?

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

- (●) No compilation errors, but undefined behavior
- ( ) No output; compiler error.
- ( ) The address value `0`
- ( ) The word "`nullptr`"

Incorrect

Question 20

0 / 1 pts

How can we print the address where ***n*** is located in memory?

```
int n{500};
```

- (●) `cout << n& << endl;`
- ( ) `cout << *(&n) << endl;`
- ( ) `cout << n << endl;`
- ( ) `cout << *n << endl;`
- ( ) `cout << &n << endl;`

Question 21

1 / 1 pts

Array subscripts are not range checked

- (●) True
- ( ) False

Question 22

1 / 1 pts

The elements of a C++ ***string*** array with no explicit initialization, created in a function will be set to ***null***.

- ( ) True
- (●) False

Question 23

1 / 1 pts

What prints?

```
int a[] = {1, 3, 5, 7, 9};
int *p = a;
cout << *++p;
cout << *p << endl;
```

- ( ) 13
- (●) 33
- ( ) 12
- ( ) 22
- ( ) None of these

Question 24

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

Question 25

1 / 1 pts

What is printed when you run this code?

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

- None of these
- No compilation errors, but undefined behavior when run
- The value 0 (stored in n)
- The address value where ***n*** is stored
- Will not compile