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 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 6:07am

Question 1

1 / 1 pts

What statement is used to signal other parts for your program that a particular error has occurred?

- `return`
- **`throw`**
- None of these
- `try`
- `raise`
- `catch`

Question 2

1 / 1 pts

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

- **True**
- False

Incorrect

Question 3

0 / 1 pts

Suppose you have written a non-interactive program that inputs data from a file. If the input file does not exist when the program executes, then you should choose which option?

- Log the error and continue
- **Throw an exception**
- Ignore the error and continue
- Terminate the program with an error message

Question 4

1 / 1 pts

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

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

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

- It does not compile.

Question 5 1 / 1 pts

A `catch` block may handle exception classes, as well as errors where `int` or `string` are thrown.

- (•) True
- ( ) False

Question 6 1 / 1 pts

The preprocessor operates on code **before** it has been compiled (ie. while it is still at the textual level)

- (•) True
- ( ) False

Question 7 1 / 1 pts

What happens when this code fragment compiles and runs?

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

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

Question 8 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
- [x] In main, y prints 4.5
- [ ] In main, y prints 4
- [ ] This code does not compile.
- [ ] In main, x prints 46

Question 9 1 / 1 pts

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

- ( ) True
- (•) False

Question 10

1 / 1 pts

Which statement is false? The elements in a vector are:

- accessed by using an index or subscript
- stored next to each other in memory
- homogeneous
- all of the same type
- None of these

Question 11

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

Question 12

1 / 1 pts

The standard library types such as `string` and `vector` are ***structured*** data types.

- True
- False

Question 13

1 / 1 pts

Which line of code can be added to print the value 4?

```
int main()
{
    struct S {int a, b; };
    vector<S> v;
    S s{3, 4};
    v.push_back(s);
    // Add code here
}
```

- `cout << v.at(0).b << endl;`
- `cout << v[0][0] << endl;`
- None of these
- `cout << v.b.at(0) << endl;`
- `cout << v.b << endl;`

Question 14

1 / 1 pts

What is the correct prototype for the output operator?

```
enum class Suit
{
   HEARTS, SPADES, CLUBS, DIAMONDS
};
```

- `ostream& operator<<(ostream& out, Suit suit);`
- `ostream& operator<<(ostream& out, Suit& suit);`
- `ostream& operator>>(ostream& out, Suit& suit);`
- `ostream& operator<<(ostream& out, const Suit& suit);`

Question 15

1 / 1 pts

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

```
struct R { int a, b; } a, b;
struct Q { int a, b; } c, d;
```

- (●) `c = d;`
- ( ) `a.b = c;`
- ( ) `a = d;`
- ( ) `b = c;`
- ( ) None of these are correct

Question 16

1 / 1 pts

Examine the following code:

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

What does x represent?

- ( ) None of these
- ( ) `true` if the value was erased and `false` otherwise
- ( ) The code does not compile because `erase()` is a `void` function
- ( ) The value that is erased (that is **1**)
- (●) An iterator that now refers to **2**

Question 17

1 / 1 pts

Assuming the following variable definition, which statement creates an object which refers to a position immediately following the last element in **v** and which allows you to change the elements in **v**?

```
vector<double> v{1.2, 2.3, 3.4};
```

- ( ) `auto c = cbegin(v);`
- (●) `auto b = end(v);`
- ( ) None of these
- ( ) `auto d = cend(v);`
- ( ) `auto a = begin(v);`

Question 18

1 / 1 pts

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

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

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

Question 19

1 / 1 pts

In C++ initializing an array with the contents of another is permitted.

- ( ) True
- (●) False

Question 20

1 / 1 pts

What prints?

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

```
cout << *p << endl;
```

- 13
- 33
- 22
- None of these
- 12

Question 21

1 / 1 pts

If ***p*** is a pointer to a structure, and the structure contains a data member ***x***, you can access the data member by using the notation: ****p.x***

- True
- False

Question 22

1 / 1 pts

Which array definition produces ***{1, 2, 0}***?

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

- a2
- None of these
- a5
- a1
- a3

Question 23

1 / 1 pts

Which line below points ***ppi*** to ***pi***?

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

- ppi = π
- *ppi = pi;
- ppi& = pi;
- None of these
- *ppi = π

Question 24

1 / 1 pts

What is the equivalent ***address-offset notation***?

```
int a[] = {1, 2, 3, 4, 5, 6, 7};
int *p = a;

cout << a[1] * 2 << endl;
```

- p + 1 * 2
- *(p + 1) * 2
- None of these
- (*p + 1) * 2
- *p + 1 * 2

Question 25

1 / 1 pts

What is printed when you run this code?

```
int num = 0;
int *ptr = &num;
num = 5;
*ptr += 5;
cout << num << " " << *ptr << endl;
```

- Undefined; none of these
- 10 5
- **10 10**
- 5 10
- 5 5