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 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 12 at 4:17pm

Question 1

1 / 1 pts

The standard library version of `stoi("UB-40")` throws a runtime exception because there is no viable conversion.

- (•) True
- ( ) False

Question 2

1 / 1 pts

What preprocessor directive ***is not used*** when you wish to create blocks of code that are only compiled under certain circumstances?

- ( ) `#ifndef`
- ( ) `#define`
- ( ) `#ifdef`
- (•) All of these may be used
- ( ) `#if`

Incorrect

Question 3

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;
}
```

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

Question 4

1 / 1 pts

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

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

- It does not compile.
- It throws a runtime exception
- It sets an error state in `in`.
- `n` is set to `1`
- None of these
- It compiles, but fails to link

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

Which of the following loop patterns are used here?

```
string s{"hello CS 150"};
for (auto e : s)
{
    if (toupper(e))
        out.put('x');
}
```

- data loop
- inline test
- limit loop
- loop-and-a-half
- counter-controlled loop
- iterator or range loop
- sentinel loop
- primed loop

Question 7

1 / 1 pts

Functions with generic parameters may use the keyword `class` or the keyword `typename` for their type parameters.

- True
- False

Question 8

1 / 1 pts

What prints?

```
string s("hello");
try {
    if (s.size() > 2) throw 42;
    if (islower(s.back())) throw "goodbye";
    if (s == "hello") throw string("hello");
    s.at(s.size()) = 'x';
    cout << "one\n";
}
catch (const int& e) { cout << "two\n"; }
catch (const string& e) { cout << "three\n"; }
catch (exception& e)  { cout << "four\n"; }
catch (...) { cout << "five\n"; }
```

- five
- three
- Undefined
- one
- four
- two

Question 9

1 / 1 pts

Assume `s1` and `s2` are C++ `string` objects. Which of these calls is illegal?

```
template <typename T>
void addem(T a, T b)
{
    cout << a << " + " << b << "->"
        << (a + b) << endl;
}
```

- `addem(4.5, 5.5);`
- `addem(s1, s2);`
- None of these
- **`addem(1.5, 2);`**
- `addem(3, 4)`

Question 10

1 / 1 pts

Which of these is the correct syntax for a lambda which returns **`true`** if a number is evenly divisible by 4?

- `[](int e) { return e % 4; }`
- None of these
- **`[](int n) { return n % 4 == 0; }`**
- `[int n] { return n % 4 == 0; }`
- `[](int e) { return e % 4 == 4; }`

Incorrect

Question 11

0 / 1 pts

Assume `vector<double> speed(5);` Which line throws a ***runtime error***?

- **`cout << speed[speed.size()];`**
- `speed[0] = speed.back()`
- `speed.erase(speed.begin());`
- None of these
- `speed.front() = 12;`

Incorrect

Question 12

0 / 1 pts

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

- **True**
- False

Incorrect

Question 13

0 / 1 pts

Examine the following code. Which element is erased?

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

- **Does not compile**
- Compiles but no element is erased
- 3
- 2
- 1

Question 14

1 / 1 pts

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

- `vector<double> salaries[10];`

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

Question 15

1 / 1 pts

The `clear()` member function removes all the elements from a `vector`.

- True
- False

Question 16

1 / 1 pts

Which of these are true?

```
int main()
{
    vector<int> v{1, 2, 3};
    for (auto i = v.size() - 1; i >= 0; i--)
        cout << v.at(i) << " ";
    cout << endl;
}
```

- Endless loop (may crash, but not necessarily)
- Compiler error (does not compile)
- Issues a compiler warning, but no error
- Crashes when run
- Prints `3 2 1`

Question 17

1 / 1 pts

Which line prints 3?

```
int main()
{
    vector<int> v{1, 2, 3};
    auto size = v.size();

    cout << v.back() << endl;       // 1.
    cout << v.front() << endl;      // 2.
    cout << v.at(0) << endl;        // 3.
    cout << v.at(size) << endl;     // 4.
    cout << v.pop_back() << endl;   // 5.
}
```

- 2
- 4
- 5
- 1
- 3

Incorrect

Question 18

0 / 1 pts

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

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

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

Question 19

1 / 1 pts

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

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

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

Question 20

1 / 1 pts

Array subscripts are not range checked

- True
- False

Incorrect

Question 21

0 / 1 pts

Examine this version of the ***swap()*** function. How do you call it?

```
void swap(int& x, int * y)
{
    . . .
}
. . .
int a = 3, b = 7;
// What goes here ?
```

- swap(a, b);
- swap(&a, b);
- swap(&a, &b);
- swap(a, &b);
- None of these

Question 22

1 / 1 pts

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

- True
- False

Incorrect

Question 23

0 / 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 24

1 / 1 pts

The ***reinterpret_cast*** instruction changes way that a pointer's indirect value is interpreted.

- True
- False

Incorrect

Question 25

0 / 1 pts

The ***static_cast*** instruction changes way that a pointer's indirect value is interpreted.

- True

False