Q-14

Due No due date **Points** 10 **Questions** 13 **Time Limit** 30 Minutes **Allowed Attempts** Unlimited

Take the Quiz Again

Attempt History

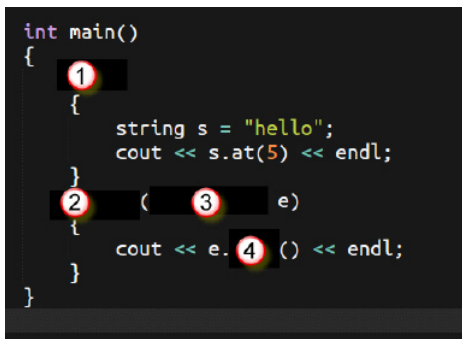
	Attempt	Time	Score
KEPT	Attempt 2	6 minutes	10 out of 10
LATEST	Attempt 3	7 minutes	9 out of 10
	Attempt 2	6 minutes	10 out of 10
	Attempt 1	9 minutes	9 out of 10

Submitted Jul 22 at 10pm

Question 1

1 / 1 pts

What is correct for # 4?



- None of these
- exception&
- while
- try
- catch
- if

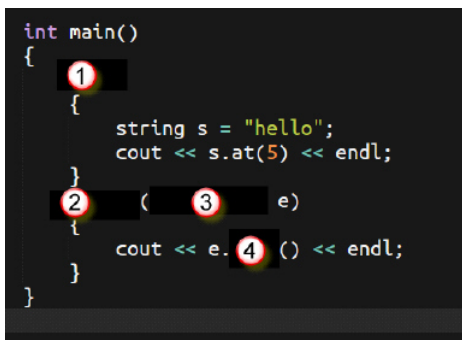
Correct!

- what

Question 2

1 / 1 pts

What is correct for # 3?



Correct!

- exception&
- what
- None of these
- catch
- try
- while
- if

Question 3

1 / 1 pts

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

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

Correct!

- `catch(...){ }`

Question 4

1 / 1 pts

What prints?

```
string s("hello");
try {
    auto x = s.at(s.size());
    cout << "one" << endl;
}
catch (const string& e) { cout << "two\n"; }
catch (exception& e)    { cout << "three\n"; }
catch (...)             { cout << "four\n"; }
```

- ○ Undefined
- Correct! ◉ three
- ○ one
- ○ two
- ○ four

Question 5

0.5 / 0.5 pts

A `catch` block may only handle objects from classes derived from `exception` or `logic_error`.

- ○ True
- Correct! ◉ False

Question 6

0.5 / 0.5 pts

A `try` block is a block of code where runtime or logical errors may occur.

- Correct! ◉ True
- ○ False

Question 7

0.5 / 0.5 pts

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

- Correct! ◉ True
- ○ False

Question 8

0.5 / 0.5 pts

A `catch` block specifies the type of exception it can catch and immediately terminates the program.

- ○ True
- Correct! ◉ False

Question 9

0 / 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;
}
```

- You Answered ◉ None of these
- Correct Answer ○ `addem(1.5, 2);`
- ○ `addem(s1, s2);`
- ○ `addem(4.5, 5.5);`
- ○ `addem(3, 4)`

Question 10

1 / 1 pts

Which call below produces 5?

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

- addem<double>(3, 2.5);
- None of these
- addem(3, 2.5);
- Correct! addem<int>(3, 2.5);
- addem(3.0, 2.5)

Question 11

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;
}
```

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

Question 12

0.5 / 0.5 pts

A function template may be defined in a header file.

- Correct! True
- False

Question 13

0.5 / 0.5 pts

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

- True
- Correct! False