

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	<a href="#">Attempt 4</a>	10 minutes	10 out of 10
LATEST	<a href="#">Attempt 4</a>	10 minutes	10 out of 10
	<a href="#">Attempt 3</a>	7 minutes	9 out of 10
	<a href="#">Attempt 2</a>	6 minutes	10 out of 10
	<a href="#">Attempt 1</a>	9 minutes	9 out of 10

Submitted Jul 22 at 10:12pm

Question 1

1 / 1 pts

What is correct for # 4?

```
int main()
{
    1
    {
        string s = "hello";
        cout << s.at(5) << endl;
    }
    2 ( 3 e)
    {
        cout << e. 4 () << endl;
    }
}
```

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

Correct!



Question 2

1 / 1 pts

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

- ☒ catch (...) { }
- ☐ catch(int x){ }
- ☐ catch(exception){ }
- ☐ catch(str) { }

Correct!

Question 3

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"; }
```

- ☐ two
- ☐ Undefined
- ☒ three
- ☐ four
- ☐ one

Correct!

Question 4

1 / 1 pts

What header file to you need to include to use the standard C++ error-handling classes?

<except>

<cstdlib>

<stdexcept>

<stderror>

<cerror>

None of these

Question 5

0.5 / 0.5 pts

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

True

False

Question 6

0.5 / 0.5 pts

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

True

False

Question 7

0.5 / 0.5 pts

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

True

False

Question 8

0.5 / 0.5 pts

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

True

False

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, U b)
{
    cout << a << " + " << b << "->"
    << (a + b) << endl;
}
```

addem(1.5, 2);

addem(s1, s2);

None of these

addem(3, 4)

addem(4.5, 5.5);

Question 10

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(3, 4)

addem(s1, s2);

Correct!

addem(1.5, 2);

addem(4.5, 5.5);

None of these

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

Correct!

In main, x prints 46.5

This code does not compile.

In main, x prints 46

Correct!

In main, y prints 4.5

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(3.5) is known as implicit instantiation.

Correct!

True

False