Q-16

Due No due date **Points** 15 **Questions** 17 **Time Limit** 30 Minutes **Allowed Attempts** Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	18 minutes	14.67 out of 15
LATEST	Attempt 4	13 minutes	12.33 out of 15
	Attempt 3	22 minutes	13 out of 15
	Attempt 2	18 minutes	14.67 out of 15
	Attempt 1	22 minutes	10.33 out of 15

Correct answers are hidden.

Submitted Jul 21 at 9:28am

Question 1

0.5 / 0.5 pts

The `push_back` member function adds elements to the end of a `vector`.

- (●) True
- ( ) False

Question 2

0.5 / 0.5 pts

Contiguous allocation means that the elements are stored next to each other in memory.

- (●) True
- ( ) False

Question 3

1 / 1 pts

What is stored in `data` after this runs?

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

- ( ) `[1, 2, 3, 0]`
- (●) `[1, 2, 3]`
- ( ) None of these
- ( ) `[2, 3]`
- ( ) `[1, 2]`
- ( ) `[]`

Incorrect

Question 4

0 / 1 pts

The following code is logically correct. What is the semantically correct prototype for `mystery()`?

```
vector<double> v{1, 2, 3};
mystery(v);
```

- ( ) Either `mystery(const vector<int>&);` or `mystery(vector<int>&);` could be correct.
- ( ) `void mystery(vector&);`
- (●) `void mystery(vector<int>&);`
- ( ) `void mystery(vector<int>);`
- ( ) `void mystery(const vector<int>&);`

Question 5

1 / 1 pts

Which lines have an identical effect?

```
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.
}
```

- 1 and 4
- 1 and 5
- None of these combinations.
- **2 and 3**
- 2 and 4

Question 6

1 / 1 pts

What prints?

```
vector<int> v{1, 2, 3, 4, 5};
cout << v.pop_back() << endl;
```

- 5
- 1
- Nothing; run-time error.
- **Nothing; compile-time error.**
- 4

Question 7

1 / 1 pts

What is stored in `data` after this runs?

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

- None of these
- [1, 2, 3]
- [2, 3]
- [1, 2, 3, 0]
- [1, 2]
- **[]**

Question 8

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

Question 9

1 / 1 pts

Which of these are true?

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

```
}
```

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

Partial

Question 10

0.33 / 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[i] << " ";
    cout << endl;
}
```

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

Incorrect

Question 11

0 / 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 prohibits you from changing **v**?

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

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

Question 12

1 / 1 pts

Assuming the following variable definition, which statement creates an object which refers to the first element in **v** and which prohibits you from changing **v**?

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

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

Question 13

1 / 1 pts

Assuming the following variable definition, which statement creates an object which refers to the first element in **v** and which prohibits you from changing **v**?

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

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

Question 14

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 prohibits you from changing **v**?

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

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

Question 15

1 / 1 pts

Examine the following code. Which element is erased?

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

- All the elements are erased
- 1
- 3
- Does not compile
- 2

Question 16

0.5 / 0.5 pts

Assume the `vector` `v` contains `[1, 2, 3]`. `v.erase(v.begin());` changes `v` to `[2, 3]`.

- True
- False

Question 17

0.5 / 0.5 pts

The statement `v.insert(v.end(), 3)` appends the element `3` to the end of the `vector` `v`.

- True
- False