Q-15

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

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	9 minutes	7.5 out of 10

Correct answers are hidden.

Submitted Jul 21 at 8:49am

Question 1

1 / 1 pts

The following is legal. Which is the correct way to access a data member in the `Rectangle` variable named `r`?

```
struct Rectangle { int length, width; };
```

- `r[0]`
- **`r.length`**
- None of these are correct
- Either `r.length` or `r->length` will work
- `r{length}`
- `r->length`

Question 2

1 / 1 pts

Examine the following code (which is legal). Which statement is correct?

```
struct Rectangle { int length, width; };
```

- `Rectangle r();`
- **`Rectangle r;`**
- None of these are correct
- `r Rectangle;`
- `Rectangle r = new Rectangle();`

Question 3

1 / 1 pts

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

```
struct Rectangle { int length, width; } big, small;
```

- None of these are correct
- `if (big == small) . . .`
- `if (big != small) . . .`
- **`if (big.length == small.width) . . .`**
- `if (big.length == width) . . .`

Question 4

1 / 1 pts

Examine the following definition. What is the syntax error?

```
struct Employee
{
   long empID;
   std::string lastName;
   double salary;
}
```

- use of `std::` in front of the `string` definition.
- **missing a semicolon after the structure definition**
- there is no syntax error
- data members cannot be different types

Question 5

1 / 1 pts

Given the following structure and variable definitions, which data members are ***default initialized***?

```
struct Employee
{
   long empID;
   std::string lastName;
   double salary;
   int age;
};

Employee bob{777, "Zimmerman", 5000000.0, 76};
```

- ○ lastName
- ○ empID
- ○ age
- ◉ None of these
- ○ salary

Incorrect

Question 6

0 / 1 pts

Given the following structure and variable definitions, which data members are ***uninitialized***?

```
struct Employee
{
   long empID{0};
   std::string lastName;
   double salary{0};
   int age = 0;
};

Employee bob;
```

- ◉ lastName
- ○ age
- ○ empID
- ○ None of them (does not compile)
- ○ None of them (compiles)
- ○ salary

Incorrect

Question 7

0 / 1 pts

Given the following structure and variable definitions, which data members are ***uninitialized***?

```
struct Employee
{
   long empID;
   std::string lastName;
   double salary;
   int age;
};

Employee bob;
```

- ☑ None of these
- ☐ salary
- ☐ lastName
- ☐ empID
- ☐ age

Question 8

0.5 / 0.5 pts

The C++ specific term for a collection of variables that have distinct names and types is a ***structure***.

- ◉ True
- ○ False

Question 9

0.5 / 0.5 pts

The following code is ***legal***.

```
struct {int hours, seconds; } MIDNIGHT{0, 0};
```

- (●) True
- ( ) False

Question 10 0.5 / 0.5 pts

A structure definition creates a new variable.

- ( ) True
- (●) False

Incorrect

Question 11 0 / 0.5 pts

The general Computer Science term for a collection of variables that have distinct names and types is a ***structure***.

- (●) True
- ( ) False

Question 12 0.5 / 0.5 pts

Structures data members must all be of the same type.

- ( ) True
- (●) False

Question 13 0.5 / 0.5 pts

It is ***illegal*** to include the same `struct` definition multiple times, even if the definitions are exactly the same.

- (●) True
- ( ) False