Week 4 Quiz
Graded Quiz • 1h

Due Nov 16, 1:59 AM EST

Week 4 Quiz

TOTAL POINTS 10

1. Which one of the following is NOT true?

1 point

○ C++ allows a variable to be declared in a user-defined member function of a user-defined class that can be defined when the function is called.

○ C++ allows a local variable to be declared in main() with an unknown type that can be defined when the program is executed.

○ C++ allows a variable to be declared in a user-defined function with an unknown type that can be defined when the function is called.

○ C++ allows a member variable to be declared in a user-defined class with an unknown type that can by defined when an object of that class is created.

2. Suppose you want to create a vector of integers. Which of the following creates an instance of the std::vector class that can contain integers?

1 point

○ int *v;

○ std::vector<int> v;

○ int v[256];

○ int<std::vector> v;

3. Which of the following will generate an error at compile time?

1 point

○ std::vector<double> v;

○ std::vector<std::vector<int>> v;

○ std::vector<char[256]> v;

○ std::vector v;

4.

```
template <typename Type>
Type max(Type a, Type b) {
    return (a > b) ? a : b;
}
```

1 point

Which one of the following exampled is a proper way to call the max function declared above in template form?

○ max<double>(5.0,10.0)

○ <Type = double>max(5.0,10.0)

○ max(5.0,10.0)

○ max<Type = double>(5.0,10.0)

5.

1 point

```
template <typename Type>
Type max(Type a, Type b) {
    return (a > b) ? a : b;
}
```

```
 6    class Just_a_double {
 7    public:
 8        double num;
 9    };
10
11    int main() {
12        Just_a_double a,b;
13        a.num = 5.0;
14        b.num = 10.0;
15        ...
16    }
```

Given the above code, which one of the expressions below, if used at line 15, will compile and not generate a compile error?

○ max("five",10.0)

○ max(a.num,b.num)

○ max(a,b)

○ max(a,10.0)

6. Which one of the following properly declares the class RubikCube derived from the base class Cube?

1 point

○ class Cube(RubikCube) {...};

○ class Cube : public RubikCube {...};

○ class RubikCube(Cube) {...};

○ class RubikCube : public Cube {...};

7.

1 point

```
class Pair {
public:
    double a,b;
    Pair(double x, double y) { a = x; b = y; }
};
```

If a class equalPair is derived from the above base class (but specializes it by adding a single boolean "isequal" member variable) then which one of the options below is a proper declaration of a constructor for equalPair?

(As a side note: Although the member variables are of type double, for the sake of this question, we are not concerned about making approximate comparisons of floating-point types, only exact comparisons. Usually, in practical usage, when you compare floating-point values, you should write a function for *approximate* comparison. That is, you should allow numbers to be considered equal if they have a very small absolute difference, even if they are not exactly the same.)

○

```
equalPair(double a, double b) {
    this->Pair(a,b);
    isequal = (a == b);
}
```

○

```
equalPair(double a, double b) : Pair(a,b) {
    isequal = (a == b);
}
```

○

```
equalPair(double a, double b) {
    isequal = (a == b);
}
```

○

```
equalPair(double a, double b) {
    Pair(a,b);
    isequal = (a == b);
}
```

8.

```
class Pair {
private:
    double a,b;
};

class equalPair : public Pair {
private:
    bool isequal;
public:
    int status();
}
```

1 point

When the function status() is implemented, which variables will it have access to?

○ Both the member variables a,b or Pair and isequal of equalPair.

○ Just the member variables a,b of Pair.

○ Just the member variable isequal of equalPair.

○ No member variables of either equalPair or Pair.

9.

```
1   class Just_a_double {
2   public:
3       double a;
4
5       Just_a_double(double x) : a(x) { }
6       Just_a_double() : Just_a_double(0) { }
7   }
```

1 point

Which constructors, if any, compile properly?

○ Both constructors on lines 5 and 6 result in compiler errors.

○ Both constructors on lines 5 and 6 compile properly

○ The constructor on line 5 compiles properly, but the constructor on line 6 results in a compiler error.

○ The constructor on line 5 results in a compiler error but the constructor on line 6 compiles properly,

10. C++ is ...

1 point

○ ... a great language for programming data structures.

○ ... the greatest language for programming data structures ever!

○ ... meh.

□ I, **Jiarong Yang**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Save

Submit