Instructor Solution

C++ Week Two Quiz

Q1: Which constructor is called in main below?

```
class Bar{

        int x_ = 0;

public:

        Bar (int a): x_(a) {}

        Bar () = default;

        Bar (const Bar& n);

        Bar(const std::initializer_list<int>& a);

};

int main(){

Bar a {0};
```
← calls the std::initializer_list constructor.
```
}
```

Q2: Create a class foo that can't be copied below:

```
class foo{
public:
    foo(const foo& n) = delete;
    foo& operator=(const foo& n) = delete;
};
```

Q3: How many destructors can a type have?

Only One.

Q4: What would be needed below for Foo to be following the RAII principals?

```
struct Foo15{

int* v_;

Foo(): v_(new int[15]) {} //acquire 60 bytes of memory

};
```

A Destructor to free the memory

~Foo() { delete [] v_; }

Q5: How would I declare a post increment ++ operator below to go to the next element:

```
class Ptr{

        int* ptr_;

public:

        Ptr(int* ptr): ptr_(ptr) {}
```

int* operator++(int){ auto tmp = ptr_; ptr_ +=1; return ptr; }

```
};
```

Q6: What three things does the keyword override accomplish?

1. Protects against typos 2. Makes it clear a function is virtual

3. Clarifies programmers intent.

Q7: Why is braced initialization preferred below:

`int a {2.3};` Storing a double into an int is a narrowing conversion. When using braced initialization this is an error.

Q8: Why would an abstract class that can't be instantiated be useful?

As an interface for a team of developers to use, that will ensure code will work togther. For polymorphism.

Q9: What exception class do all exceptions in the standard library inherit from?

std::exception

Q10: What happens to an uncaught exception?

The program calls std::terminate

Q11: What is the difference between a struct and a class?

A struct has a default access modifier of public while a class has a default access modifier of private.

Q12: What is a traditional Error Handling method that can be used if you can't use exceptions?

Returning an error code

Q13: Why must we ensure that no exceptions are thrown in our exception handlers?

Program will call std::terminate

Q14: When should a developer defined a copy assignment operator for a class? Rule of Five

When he defines a copy constructor or a move constructor or assignment, or if he defines a destrutor.