

ECS 34 Midterm

Geoffrey Mohn

TOTAL POINTS

56 / 100

QUESTION 1

1 Q1 10 / 10

- ✓ - **0 pts** Correct
- **8 pts** Blank
- **6 pts** Missing or extra curly brackets
- **3 pts** Missing semicolon

QUESTION 2

2 Q2 0 / 10

- **0 pts** Correct
- **8 pts** Blank
- **5 pts** !i executed on 0
- ✓ - **10 pts** Incorrect, should be 3

QUESTION 3

3 Q3 10 / 10

- ✓ - **0 pts** Correct
- **8 pts** Click here to replace this description.
- **5 pts** Missed issued of memory leak
- **10 pts** Incorrect (weak pointers break cycle, prevent memory leak)

QUESTION 4

4 Q4 10 / 10

- ✓ - **0 pts** Correct
- **8 pts** Blank
- **10 pts** Incorrect, should be "D = 3.25"

QUESTION 5

5 Q5 0 / 10

- **0 pts** Correct
- **8 pts** Blank
- ✓ - **5 pts** Incorrect issue, cannot modify in const function
- ✓ - **5 pts** Incorrect solution, should remove const

QUESTION 6

6 Q6 10 / 10

- ✓ - **0 pts** Correct
- **8 pts** Blank
- **10 pts** Incorrect, describing issue of memory leak
- **5 pts** No or incorrect explanation
- **10 pts** Incorrect

QUESTION 7

7 Q7 3 / 10

- **0 pts** Correct
- **8 pts** Blank
- **3 pts** Incorrect, A[0] => Stack
- ✓ - **3 pts** Incorrect B[1] => Heap
- ✓ - **4 pts** Incorrect B => Stack

QUESTION 8

8 Q8 10 / 10

- ✓ - **0 pts** Correct
- **8 pts** Blank
- **10 pts** Incorrect, class is private by default, struct is public by default

QUESTION 9

9 Q9 3 / 10

- **0 pts** Correct
- **8 pts** Blank
- ✓ - **4 pts** Missed int constructor
- **3 pts** Missed operator=
- ✓ - **3 pts** Missed /

QUESTION 10

10 Q10 0 / 10

- **0 pts** Correct
- **8 pts** Blank
- **5 pts** Partially incorrect

✓ - **10 pts** Incorrect, bool is a template specialization

QUESTION 11

11 Notes 0 / 0

✓ - **0 pts** Correct

- **0.5 pts** Wrong name location


- **1 pts** Name not typed

- **2 pts** Wrong paper

- **2 pts** No notes

Midterm

I shall not cheat on, or knowingly give or receive assistance on this examination, and I shall not condone cheating by other persons. I shall not sit next to anyone (even if an empty seat is between us) that I am partnering with or have partnered with on any assignment, study with or have studied with regularly, or routinely spend time with socially. I understand that if I am suspected of academic misconduct that the Office of Student Judicial Affairs will handle the matter. I understand if I am found to be guilty of academic misconduct regarding this examination, or if I violate the rules on whom I sit next to, that I may receive a score of zero. If permitted to use handwritten notes, I agree to submit my notes when turning in this examination. I understand that if I leave a problem blank I will receive 20% of the credit for the problem.

Name (print): Geoffrey Mohn Student ID: 912568148
Signature:  Date: 10/30/19
Student Left: Emaan Riaz Student Right: Noah de la Cruz

Seat Number: left H 3

- 1) (10 points) Given the following C++ code, fix any syntactic errors:

```
int foo(int p){
    int j = 4;

    for(int i = 0; i < p; i++){
        j += i;
    }

    return j;
}
```

- 2) (10 points) Given the following C++ code, what value will be in k at the end of the block?

```
{
    int j = 12, k = 3;
    for(int i = 0; i < 5; i++){
        if(!i){
            k = 0;
        }
        else if(i < 2){
            k *= 3;
        }
        else{
            j = 6;
            k++;
        }
    }
}

k = 21
```

if not i when i is 0
i = 2 3 4
or 3

- 3) (10 points) Shared pointers keep track of reference counts of an object so it is clear when the object can be deleted. What is the purpose of weak pointers? Why are they even necessary?

Allows multiple references but isn't reference counted
 like Shared Ptrs. Shared Ptrs can cause memory leaks
 if circularly referenced. If we use weak pointers with
 a shared pointer the weak pointer can be freed
 because it's not ref counted

- 4) (10 points) Given the following code, what will be the output?

```

void foo(double v) {
    std::cout<<"D = "<<v<<std::endl;
}
void foo(int v) {
    std::cout<<"I = "<<v<<std::endl;
}
int main() {
    auto i = 2 + 1.25;
    foo(i);
    return 0;
}

```

$i = 3.25$

$D = 3.25 \text{ m}$

- 5) (10 points) The following code will not compile. What is the problem, and what could be done to make it compile?

```

class C1 {
    int D1;
    void foo(int i) const {
        D1 += i;
    }
}

```

$D1 = i;$

$D1$ has not been assigned a value so we can't add to it

- 6) (10 points) In C++, if shared pointers are used, is it possible for there to be a dangling reference? If so, explain a situation where this will occur. If not, explain why it is not possible for there to be dangling references.

It is not possible, because shared ptrs are ref counted. When another ptr points to the same address the ref count is incremented. When ptr stops pointing to the address the ref count is decremented. The memory is freed when the ref count = 0.

- 7) (10 points) Given the following code, at the specified line what memory section(s) are A[0], B[1], and B in? Justify your answer.

```

int main() {
    int A[] = {3, 2, 1};
    int *B = new int[2]{7, 6};
    A[0] = P[1]; // At this line
    delete [] P1; B[1]
    return 0;
}

```

A[0] is on the stack
B[1] is on the stack
B is on the heap

Name: Geetha Mohan

- 8) (10 points) The following code won't compile, but if the developer changes class to struct it will. Why will the code with C1 work as a struct, but not a class?

```
class C1{
    int D1 = 6;
public:
    C1(){};
    int D() const{
        return D1;
    };
};
```

Because class defaults D1 as Private, when trying to access D1 from main(), it isn't allowed to as it isn't in the class scope. By switching to struct, D1 becomes Public by default. This allows main to access it.

```
int main(){
    C1 O;
    O.D1++;
    return 0;
}
```

- 9) (10 points) Given the following C++ code, what functions have to be implemented by class C1?

```
int main(){
    C1 I, J{1}, K{2};
    I = J / K;
    return 0;
}
```

Operator =
C1()
~C1()
I

- 10) (10 points) The following code compiles, it also compiles for int instead of double, but when the developer changes int to bool it won't compile. Why won't it compile with bool?

```
template<typename T>
T *foo(T &v){
    return &v;
}

int main(){
    std::vector<double> Vect;
    Vect.resize(3);
    auto P = foo(Vect[0]);
    return 0;
}
```

Vec bool is single bit
unknown handling

Vector bool's memory creates unexpected functionality when passed by reference

Vector bool's memory creates unexpected handling by the compiler when resized & passed by reference