

Name: Gravin Hamille

## 1 const

1. In your own words, describe what the const keyword does.

A. Makes the declared variable or value immutable (can't change it)

2. Where are two places that you should use the const keyword?

A. ↳ TOP of program where declare global variables (literals)  
↳ In <sup>getters</sup> methods where you want the return value to be immutable

3. Mark the following code. Circle the lines you think will cause compiler errors.

```
1 class Point {  
2 public:  
3   Point(const int x, const int y) {  
4     x_ = x;  
5     y_ = y;  
6   }
```

```
7  
8   int get_x() const { return x_; }  
9
```

```
10  int get_y() const {  
11    y_++;  
12    return y_;  
13  }
```

```
14  
15 private:  
16   int x_;  
17   int y_;
```

```
18 };  
19  
20  
21
```

```
22 void CreateSquare(const int y) {  
23   y = y * y;  
24 }
```

```
25  
26 void CreateCube(const int &y) {  
27   y = y * y * y;  
28 }
```

```
29  
30 void CreateQuadruple(int &y) {  
31   y = y * y * y * y;  
32 }
```

```
33  
34 int main() {  
35   int a = 10;  
36   const int b = 7;  
37   a++;  
38   b++;  
39 }
```

int get\_x() const { return x\_; }

constant member function →

← attempting to modify const var

constant variable argument

Name:

↑ Allocates memory for the object...

(Parameterized)

• Auxiliary Constructor  
↳ Provide parameters

## 2 Constructors

↳ If you have one; make sure to include Default Constr.

1. What is a constructor?

A. A class method that creates / initializes a new instance of the class object

2. Write the equivalents of the following constructors using initialization lists.

```
1 class Point {
2 public:
3     // 0 parameter constructor initializes Point at the origin
4     // constructor 1
5     Point() {
6         x_ = 0;
7         y_ = 0;
8     }
9
10    // constructor 2
11    Point(const int x, const int y) {
12        x_ = x;
13        y_ = y;
14    }
15
16 private:
17     int x_;
18     int y_;
19 };
20
```

→  $\text{Point}() : x\_ (0), y\_ (0) \{ \}$

$\text{Point}(\text{const int } x, \text{const int } y) : x\_ (x), y\_ (y) \{ \}$

```
21 class Library {
22 public:
23     // constructor 3
24     Library(const std::string name) {
25         name_ = name;
26         std::vector<Book> tmp;
27         shelf_ = tmp;
28     }
29
30 private:
31     std::string name_;
32     std::vector<Book> shelf_; // Book is defined elsewhere in the code for us
33 };

```

$\text{Library}(\text{const std::string name}) : \text{name\_}(\text{name}),$   
 $\text{std::vector<Book>}(\text{tmp}),$   
 $\text{shelf\_}(\text{tmp}) \{ \}$

Name:

3. Which of the constructors in the problem above are used in each of the following lines of code? How many instances of the class in question are created? (Or does the line of code cause an error?)

```
1 Point p;  
2  
3 Point p2(1, 3);  
4  
5 Point p3[500]; → Error: Type mismatch  
6  
7 Point p4(1, 2, 3); → Error: Too many args  
8  
9 Point * p5 = new Point(); -  
10  
11 Library lib; → Error missing args  
12  
13 Library * lib2 = new Library(); -  
14  
15 Library * lib3 = new Library("norlin");
```

A.

3 enum

1. Create an enum class, **Suit** to represent the suit in a deck of cards.

A.

2. Create a struct, **Card**, to represent a card in a deck of cards. You must use your **Suit** enum. You may use any other values that you choose.

A.

3. Override operator== in your Card struct.

A.

4. Design a class, **Deck**. Write down the fields and methods that this class should have. Don't implement them!

A.

## 4 Lecture 1 Review

1. What is a class? What is a struct? How are classes and structs different from one another?

A. A class defines an object, it's attributes and methods for an instance of that obj.

• A struct is similar to a class but works a just an organization of data, NO METHODS

2. In your own words, what is an instance of a class?

A. An instance of a class is a new creation of that class's object/structure

Name:

3. Where can you access public fields? private fields? Are fields in a struct public or private?

A. Can access public fields with an instance of an class

- Can access private fields with getter methods

~ Fields in a struct should be public

4. What are the differences between the following three variable declarations? How would you increment the value of x from y and z?

`int x = 2; int *y = &x; and int &z = x;`

• `int x` is a basic local variable

Increment  $\rightarrow x++;$

• `int *y = &x` is a pointer to the address of x

Increment  $\rightarrow$

• `int &z = x;` is a reference to the variable x

Increment  $\rightarrow$