# 1 const

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

A. The ‘const’ keyword means that the value of the variable can’t be changed or modified.

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

A. Two places that you should use the const keyword is:

1. Function parameters

2. Getter functions for variables

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

A screenshot of a social media post

Description automatically generated

1. (Cant highlight the image so I have listed the lines below)

LINES: 11, 23, 27, 38

# 2 Constructors

1. What is a constructor?

A. A constructor creates a class object

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

A screenshot of a social media post

Description automatically generated

A.

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

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

//needs variable to be in initialized in private

Point::Point(const std::string name) name\_(name){

Std::vector<Book>tmp;

shelf\_ = tmp;

}

3. Which of the constructors in the problem above are use 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?) //new used in line 9 is stored in the heap

A screenshot of a cell phone

Description automatically generated

1. ERRORS lines: 7 (no three parameter constructor) , 11 (no default const), 13 (no default const)

# 3 enum

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

A.

enum Class Suit{Hearts,Clubs,Spades,Diamonds};

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.

Struct card{

int num;

Suit s;

}

3. Override operator== in your Card struct.

A.

bool operator==(const Card &c2){

return num==c2.num && s == c2.s;

}

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

A.

class Deck {

public:

Deck();

Private:

std::vector<Card> deck\_;

private:

# 4 Lecture 1 Review

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

A.

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

A.

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

A.

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;**