Name: Felipe Lima

1 const

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

A.

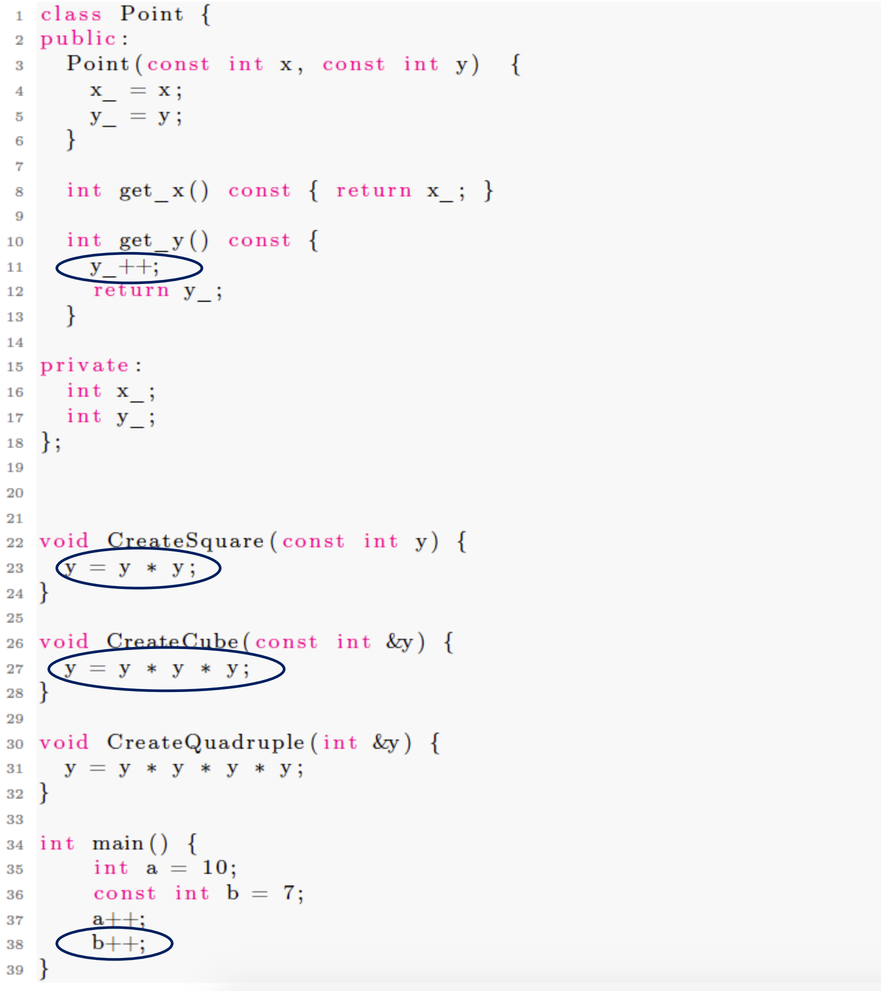
The const keyword means “unchangeable”, that is, the variable assigned to a const, cannot be changed later in the code.

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

A.

- Local variable that won’t change throughout the code.
- Getter functions that won’t change the object that is calling them.

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



```
class Point {
public:
  Point(const int x, const int y)  {
    x_ = x;
    y_ = y;
  }

  int get_x() const { return x_; }

  int get_y() const {
    y_++;
    return y_;
  }

private:
  int x_;
  int y_;
};



void CreateSquare(const int y) {
  y = y * y;
}

void CreateCube(const int &y) {
  y = y * y * y;
}

void CreateQuadruple(int &y) {
  y = y * y * y * y;
}

int main() {
    int a = 10;
    const int b = 7;
    a++;
    b++;
}
```

A.

Name: Felipe Lima

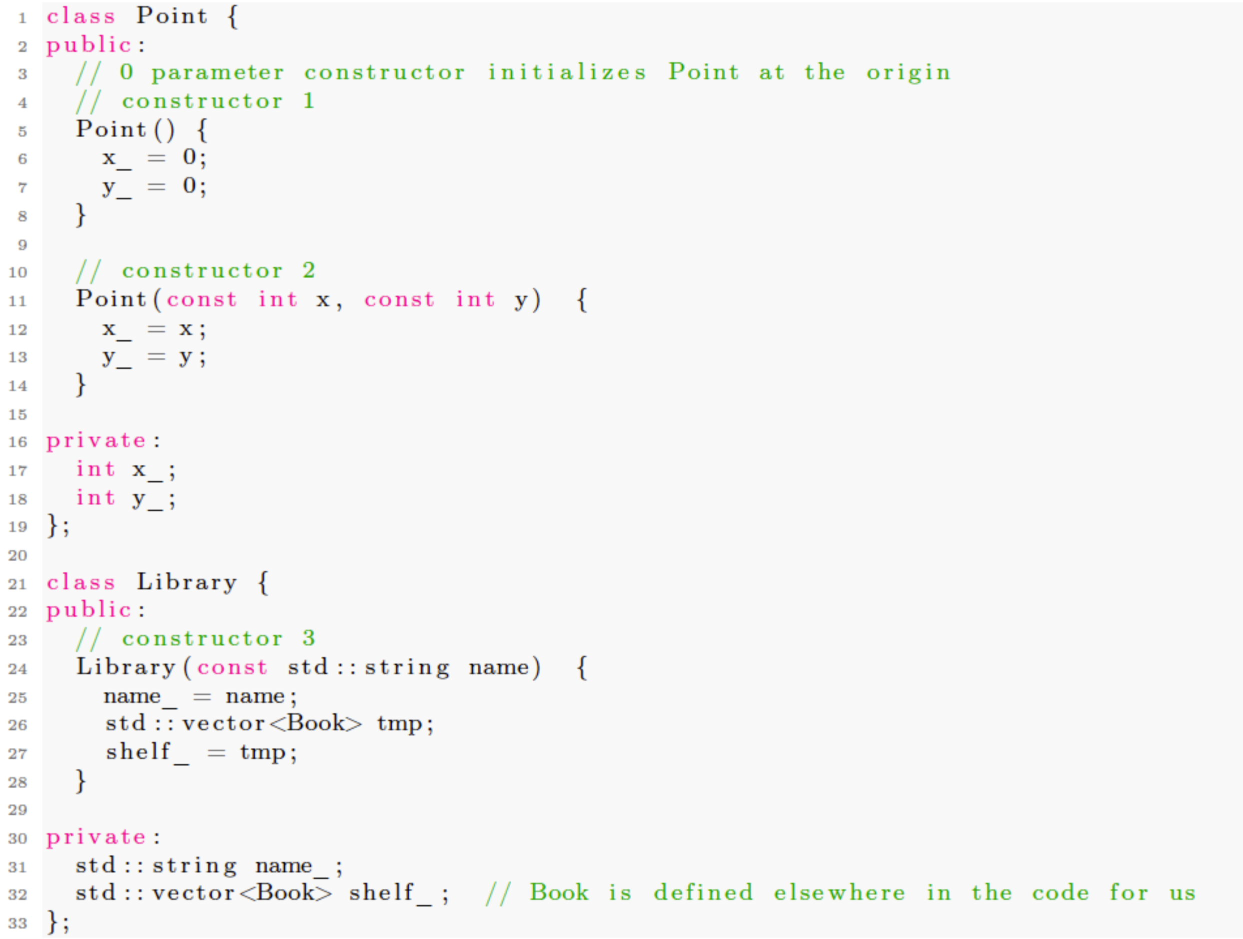
2 Constructors

1. What is a constructor?

A.

A constructor is a member function of a class that initializes objects of that class.

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



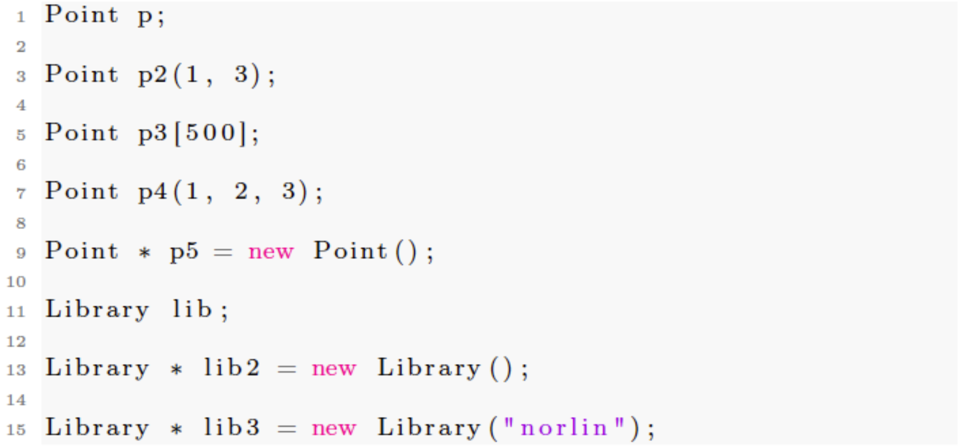
```
class Point {
public:
  // 0 parameter constructor initializes Point at the origin
  // constructor 1
  Point() {
    x_ = 0;
    y_ = 0;
  }

  // constructor 2
  Point(const int x, const int y) {
    x_ = x;
    y_ = y;
  }

private:
  int x_;
  int y_;
};

class Library {
public:
  // constructor 3
  Library(const std::string name) {
    name_ = name;
    std::vector<Book> tmp;
    shelf_ = tmp;
  }

private:
  std::string name_;
  std::vector<Book> shelf_;  // Book is defined elsewhere in the code for us
};
```

A.

```
Point::Point():x_(0),y(0){}
Point::Point(const int x, const int y):x_(x),y_(y){}
Library::Library(const, std::string name):name_(name){
        std::vector<Book> tmp;
        shelf_ = tmp;
}
```

Name: Felipe Lima

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?)



```
Point p;

Point p2(1, 3);

Point p3[500];

Point p4(1, 2, 3);

Point * p5 = new Point();

Library lib;

Library * lib2 = new Library();

Library * lib3 = new Library("norlin");
```

A.
P1: Point(); | 1
P2: Point(const int x, const int y); | 1
P3: Point(); | 500
P4 : error, no matching constructor
Library lib: error, no matching constructor
Library lib2: error, no matching constructor
Library lib3: Library(std::string name) | 1

3 enum

1. Create an enum class, Suit to represent the suit in a deck of cards.
A.
enum class Suit {Hearts, Spades, Clubs, 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;
}
```

Name: Felipe Lima

3. Override operator== in your Card struct.

A.

```
struct card{
        int num;
        Suit s;
        bool operator==(card &c1){ return num == c1.num && s == c2.num;};
}
```

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

A.

```
Class Deck{


}
```

Name: Felipe Lima

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 is a date type that’s holds its own information either public or private and its used to create objects and different instances of the same class.
Struct is a group of variables under the same name. One big difference from classes is security. Structs are not secure while classes can hide its programing details.

2. In your own words, what is an instance of an object?
A.
An object that has its own set of values for members of the class.

3. Where can you access public fields? private fields? Are fields in a struct public or private?
A.
Public: anywhere in the code.
Private: can only be accessed withing the class’s member functions.
Fields on a struct are public by default.

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;

```
x++;
(*y)++;
z++;
```