Courtney Makua

01/12/2022

CSCI 3731 A

Dr Edward Pier

1. What is wrong with the following code and how would you fix it?

```
#ifndef PROJECTILE_H
#define PROJECTILE_H
class Projectile {
private:
double position;
double velocity;
public:
Projectile(double position, double velocity);
virtual ˜Projectile();
double getPosition() const;
double getVelocity() const;
} // end of Projecile class
#endif
```

Missing a semicolon after the end curly brace.

The following is the definition of the constructor for the Projectile class above, but there are three things wrong with it. What are they and how would you fix them?

```
Projectile(int position, int velocity) {
this.position = position;
this.velocity = velocity;
} // end of constructor
```

Projectile:projectile

This->position=position;

This->velocity=velocity;

3. Describe each of the following methods

(a) int* method(int* arg); returns a pointer

(b) const int* method(int* arg); Returns a pointer to a const int

(c) const int* const method(int* arg); Returns a const int to a const int

(d) const int* const method(const int* arg); Takes a pointer to a const int and returns a pointer to a const int

(e) const int* const method(const int* arg) const; Takes a pointer to a const int, returns a pointer to a const int and does not modify the object.

What is the difference between a pointer and a reference?

A reference is automatically dereferenced, needs to be assigned when declared, cant change what it points to, and can't be nullptr.

Why would you never declare a method like this?

void method(const int value);

The int "value" is being set as const which means it can not be changed but is isn't good because the method is it being passed to could possibly try to change that value.