Christopher Aranda

10/19/18

Edward Peir

CSCI 3731

HW-07

1. (10 pts) 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 speed;

public:

Projectile(double speed, double velocity);

virtual ̃Projectile();

double getSpeed() const;

double getVelocity() const;

void set (double speed, double velocity);

} ;// end of Projectile class

#endif

2. (10 pts) 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::Projectile (int speed, int velocity) {

this->speed = speed;

this->velocity = velocity;

} // end of constructor

1. Nothing’s being done with the arguments speed and velocity.
2. Use -> pointers.
3. Indicate which class the method belongs to (To be reworded).

3. (10 pts) Describe each of the following methods

(a) int\* method(int\* arg);

// returns a reference to an int

//Both arguments are modified.  
(b) const int\* method(int\* arg);

// returns a constant reference to an int

//Both arguments not modified.  
(c) const int\* const method(int\* arg);

// returns a constant reference to an int

//Left argument is modified, right argument is not.

(d) const int\* const method(const int\* arg);

// returns a constant reference to an int

//Left argument is modified, but the right argument is not  
(e) const int\* const method(const int\* arg) const;

// returns a constant reference to an int

//Both arguments are not modified.

1. In what ways are C++ strings better than C strings? In what ways are C strings better than C++ strings?
   1. C strings are char arrays that end with null. For C strings, it’s the programmer’s responsibility to manage the memory of the char arrays while modifying the strings. C++ strings are objects containing the automated memory management, making it easier for the programmer. C++ strings are immutable, but can be declared const.
2. What is the difference between a pointer and a reference?
   1. References are automatically dereferenced, whereas pointers need to be dereferenced when necessary. Pointers can be assigned to null, but references cannot.
3. What is a destructor for?
   1. Destructors delete heap memory allocated by the class. They destruct, or delete, objects.