HOMEWORK 6

CSCI 3731-A

Problem solving in C++

1. (10 pts) What is wrong with the following code and how would you ﬁx it?

#ifndef PROJECTILE\_H // #ifndef guards against using the file more then once

#define PROJECTILE\_H

class Projectile {

private: double position; variable position not used

double speed;

public: Projectile(double speed, double velocity) throw(); - THE VARIABLE VELOCITY WASN’T DECLARED

virtual ˜Projectile() throw();

double getSpeed() const throw();



double getVelocity() const throw(); - don’t need throw(); in the variable or on the destructor.

}; need close bracket and simi-colon

#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(int speed, int velocity) { should be Projectile::Projectile(int speed, int velocity){

this.x = x; should be this.x->x=x;

this.y = y; should be this.y->y=y;

} // end of constructor

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

(a) int\* method(int\* arg); This method receives a int pointer which can be changed and returns a pointer to an int

(b) const int\* method(int\* arg); this method receives a int pointer that can be modified ard return a const int that can not be modified.

(c) const int\* const method(int\* arg); in this case the method receives an int pointer and returns a const to a const int. because the const is before the method the return value or object is const not the method itself.

(d) const int\* const method(const int\* arg); this method takes in a const int and returns a const to a const int.

(e) const int\* const method(const int\* arg) const; this method takes in a const int and returns a const to a const int. Here the method is a const because const comes at the end.

4. In what ways are C++ strings better than C strings? In what ways are C strings better than C++ strings?

C++ strings are mutable and can be changed but Java strings are immutable and must create new strings since can’t change once created.

5. What is the difference between a pointer and a reference? Pointer is used by dereferencing the pointer how ever the reference can be used again.

6. What is a destructor for?

Destructor clean the heap memory when using an array or any = new object in c++. In Java most garbage collection happens automatically.

7. Write an Angle class. The interesting things about angles is that they cycle through the range 0◦ to 360◦. For example, 250◦ + 190◦ = 80◦, and 40◦−90◦ = 310◦. In other words, you do a math operation and then while the angle is greater than 360, you subtract 360. While it is less than 0, you add 360. Implement the following operators: • +, -, +=, and -= operators so that you can add and subtract pairs of angles. • \*, /, \*= and /= so that you can multiply and divide an angle by a double. • = so that you can assign from either another Angle or a double. • == to compare two Angles. After each of these operations, make sure the angle is between 0 and 360 degrees. Make the class printable. Write a program that tests your class.