Name: Tai Ngoc Bui
CSCI 3731 - hw07

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;

} // end of Projecile class
#endif
```

At the end of the class, there is no semi colon. Leaving it off will result with cryptic error messages from inside one or more .cc files that include the header file. We fix by adding semi colon after the last curly bracket.

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) {

this.x = x;

this.y = y;

} // end of constructor
```

The instance variables should be speed and velocity to match with what were declared in the header files. Moreover, the arguments should be double type, not int. The constructor does not include what class it belongs to. Finally, this is a pointer to an object in C++, thus it should be dereferenced first before being used. The fix should be

```
Projectile::Projectile (double speed, double velocity) {

        this -> speed = speed;

        this -> velocity = velocity;

} //end of constructor
```

Name: Tai Ngoc Bui
CSCI 3731 - hw07

3. (10 pts) Describe each of the following methods
(a) int* method(int* arg);
Declaring a method which takes into a pointer to an int as argument and return a pointer to an int
(b) const int* method(int* arg);
Declaring a method which takes into a pointer to an int as argument and return a pointer to a constant int
(c) const int* const method(int* arg);
Declaring a method which takes into a pointer to an int as argument and return a constant pointer to a constant int
(d) const int* const method(const int* arg);
Declaring a method which takes into a pointer to a constant int as argument and return a constant pointer to a constant int
(e) const int* const method(const int* arg) const;

Declaring a constant method which takes into a pointer to a constant int and return a constant pointer to a constant int.

4. In what ways are C++ strings better than C strings? In what ways are C strings better than C++ strings?
C++ strings help users to avoid computer security holes which would be resulted from using C strings. C++ strings are safer and more convenient to use than C strings. Nevertheless, C strings is a more lightweight and sufficient design compared to C++ strings.

5. What is the difference between a pointer and a reference?
References generally is similar to pointer; however, they are automatically dereferenced. References have two restrictions including they must be initialized when they are declared and after that you cannot reassign the references to point anything else. Thus, a references can never be null.

6. What is a destructor for?

The destructor gets called when a class is deleted. The main use of destructor is to delete heap data created by the object.

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.