Main.cpp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <iostream>

using namespace std;

#include "enemy.h"

#include "ninja.h"

#include "monster.h"

/\* I think the use of this method of doing this is to use the

same function name to do slightly different things \*/

int main()

{

ninja n;

monster m;

enemy \*enemy1 = &n;

enemy \*enemy2 = &m;

// if when declaring you dont put the virtual in front of enemy.attack

// then it will print "the enemy has struck" instead of the ninja/monster

// the program wont run using n -> enemy1 : idk why

enemy1-> attack();

enemy2 -> attack();

return 0;

}

enemy.h\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#ifndef ENEMY\_H

#define ENEMY\_H

class enemy

{

public:

enemy();

virtual void attack();

protected:

private:

};

#endif // ENEMY\_H

enemy.cpp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <iostream>

using namespace std;

#include "enemy.h"

#include "ninja.h"

#include "monster.h"

enemy::enemy()

{

}

void enemy::attack()

{

cout << "the enemy has struck" << endl;

}

ninja.h\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#ifndef NINJA\_H

#define NINJA\_H

class ninja: public enemy

{

public:

ninja();

void attack();

protected:

private:

};

#endif // NINJA\_H

ninja.cpp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <iostream>

using namespace std;

#include "enemy.h"

#include "ninja.h"

#include "monster.h"

ninja::ninja()

{

//ctor

}

void ninja::attack()

{

cout << "the ninja has struck" << endl;

}

monster.h\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#ifndef MONSTER\_H

#define MONSTER\_H

class monster: public enemy

{

public:

monster();

void attack();

protected:

private:

};

#endif // MONSTER\_H

monster.cpp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <iostream>

using namespace std;

#include "enemy.h"

#include "ninja.h"

#include "monster.h"

monster::monster()

{

}

void monster::attack()

{

cout << "the monster has struck" << endl;

}