main.cpp \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include "monster.h"

#include "ninja.h"

#include "nazi.h"

#include <iostream>

using namespace std;

int main()

{

// declare objects for ninja and nazi

ninja ni;

nazi na;

//since ni/na inherited all of monsters stuff this is valid

// 1. create a monster object

// 2. have that object point to the ninja/nazi object adress

monster \*monster1 = &ni;

monster \*monster2 = &na;

// now when we point to strike it changes the int variable strike

// pertenant to the ninja and nazi class

// so when monster calls the function ninjastrike/nazistrike

// the program changes the strike variable independently for

// ninja and nazi

monster1 -> strike = 17;

monster2 -> strike = 62;

ni.ninjaStrike();

na.naziStrike();

return 0;

}

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

#ifndef MONSTER\_H

#define MONSTER\_H

class monster

{

public:

monster();

int strike;

void ninjaStrike();

void naziStrike();

protected:

private:

};

#endif // MONSTER\_H

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

#include "monster.h"

#include "ninja.h"

#include "nazi.h"

#include <iostream>

using namespace std;

monster::monster()

{

//ctor

}

void monster::ninjaStrike()

{

cout<< "Ninja choOP! -" << strike << endl;

}

void monster::naziStrike()

{

cout << "Nazi garbage -" << strike << endl;

}

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

#ifndef NINJA\_H

#define NINJA\_H

class ninja: public monster

{

public:

ninja();

protected:

private:

};

#endif // NINJA\_H

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

#include "monster.h"

#include "ninja.h"

#include "nazi.h"

#include <iostream>

using namespace std;

ninja::ninja()

{

//ctor

}

nazi.h \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#ifndef NAZI\_H

#define NAZI\_H

class nazi: public monster

{

public:

nazi();

protected:

private:

};

#endif // NAZI\_H

nazi.cpp \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include "monster.h"

#include "ninja.h"

#include "nazi.h"

#include <iostream>

using namespace std;

nazi::nazi()

{

//ctor

}