#include <iostream>

#include <string>

#include <cstdlib>

#include <ctime>

// Character class to represent players and enemies

class Character {

private:

std::string name;

int health;

int attackPower;

int defense;

public:

Character(std::string n, int h, int atk, int def)

: name(n), health(h), attackPower(atk), defense(def) {}

std::string getName() const { return name; }

int getHealth() const { return health; }

int getAttackPower() const { return attackPower; }

int getDefense() const { return defense; }

void takeDamage(int damage) {

int actualDamage = damage - defense;

if (actualDamage < 0) actualDamage = 0;

health -= actualDamage;

if (health < 0) health = 0;

}

bool isAlive() const {

return health > 0;

}

void attack(Character &target) const {

std::cout << name << " attacks " << target.getName() << "!\n";

target.takeDamage(attackPower);

}

};

// Battle system

class Battle {

public:

void startBattle(Character &player, Character &enemy) {

std::cout << "A battle between " << player.getName() << " and " << enemy.getName() << " begins!\n\n";

while (player.isAlive() && enemy.isAlive()) {

player.attack(enemy);

std::cout << enemy.getName() << " has " << enemy.getHealth() << " health remaining.\n\n";

if (!enemy.isAlive()) {

std::cout << enemy.getName() << " has been defeated!\n";

break;

}

enemy.attack(player);

std::cout << player.getName() << " has " << player.getHealth() << " health remaining.\n\n";

if (!player.isAlive()) {

std::cout << player.getName() << " has been defeated!\n";

break;

}

}

if (player.isAlive()) {

std::cout << player.getName() << " is victorious!\n";

} else {

std::cout << enemy.getName() << " is victorious!\n";

}

}

};

int main() {

srand(static\_cast<unsigned int>(time(0))); // Seed for random number generation

// Create characters

Character player("Hero", 100, 25, 5);

Character enemy("Goblin", 50, 15, 2);

// Start the battle

Battle battle;

battle.startBattle(player, enemy);

return 0;

}