CS 11 Data Structures and Algorithms

Assignment 7.1: Inheritance 1

Return to Course Homepage

Assignment 7.1

```
//
// creature.h
//
#ifndef creature_h
#define creature_h

#include <string>

namespace cs_creature {

    class creature {
    private:
        int strength;           // how much damage this creature inflicts
        int hitpoints;          // how much damage this creature can sustain
        const static int DEFAULT_STRENGTH = 10;
        const static int DEFAULT_HITPOINTS = 10;

    public:
        creature();
        creature(int newStrength, int newHitpoints);
        int getDamage() const;
        std::string getSpecies() const;
        int getStrength() const;
        int getHitpoints() const;
        void set_strength(int);
        void set_hitpoints(int);
    };
}
#endif

//-------------------------------------------------------------------------------------------
//
//  creature.cpp
//


#include "creature.h"
#include <cstdlib>

using namespace std;

namespace cs_creature {

    creature::creature(){
        strength = DEFAULT_STRENGTH;
        hitpoints = DEFAULT_HITPOINTS;
    }




    creature::creature(int newStrength, int newHitpoints){
        strength = newStrength;
        hitpoints = newHitpoints;
    }




    string creature::getSpecies() const {
        return "creature";
    }




    int creature::getDamage() const {

        return (rand() % strength) + 1;
    }
```

```
    int creature::getStrength() const {
        return strength;
    }




    int creature::getHitpoints() const {
        return hitpoints;
    }




    void creature::set_strength(int newStrength){
        strength = newStrength;
    }




    void creature::set_hitpoints(int newHitpoints){
        hitpoints = newHitpoints;
    }
}
//--------------------------------------------------------------------------------------
//
//  human.h
//

#ifndef human_h
#define human_h

#include "creature.h"
#include <string>

namespace cs_creature {

    class human: public creature {
    public:
        human();
        human(int newStrength, int newHitpoints);
        int getDamage() const;
        std::string getSpecies() const;
    };
}
#endif

//--------------------------------------------------------------------------------------
//
//  human.cpp
//

#include "human.h"
#include <iostream>
using namespace std;

namespace cs_creature {

    human::human(){
    }


    human::human(int newStrength, int newHitpoints)
    : creature(newStrength, newHitpoints){
    }



    string human::getSpecies() const {
        return "human";
    }


    int human::getDamage() const {
        int damage = creature::getDamage();
        cout << "The human attacks for " << damage << " points!" << endl;
        return damage;
    }
}
//--------------------------------------------------------------------------------------
//
//  elf.h
//

#ifndef elf_h
#define elf_h
```

```
#include "creature.h"
#include <string>

namespace cs_creature {

    class elf: public creature {
    public:
        elf();
        elf(int newStrength, int newHitpoints);
        int getDamage() const;
        std::string getSpecies() const;
    private:
        static const double MAGICAL_ATTACK_PROBABILITY;
    };
}
#endif

//-----------------------------------------------------------------------------
//
//  elf.cpp
//

#include "elf.h"
#include <iostream>
using namespace std;

namespace cs_creature {

    const double elf::MAGICAL_ATTACK_PROBABILITY = 0.5;

    elf::elf(){
    }




    elf::elf(int newStrength, int newHitpoints)
    : creature(newStrength, newHitpoints){
    }




    string elf::getSpecies() const {
        return "elf";
    }




    int elf::getDamage() const {
        int damage = creature::getDamage();
        cout << "The elf attacks for " << damage << " points!" << endl;

        if (rand() % 100 * 0.01 < MAGICAL_ATTACK_PROBABILITY) {
            cout << "Magical attack inflicts " << damage << " additional damage points!" << endl;
            damage = damage * 2;
        }
        return damage;
    }
}
//-----------------------------------------------------------------------------
//
//  demon.h
//

#ifndef demon_h
#define demon_h

#include "creature.h"
#include <string>

namespace cs_creature {

    class demon: public creature {
    public:
        demon();
        demon(int newStrength, int newHitpoints);
        int getDamage() const;
        std::string getSpecies() const;
    private:
        static const int DEMONIC_ATTACK_DAMAGE = 50;
        static const double DEMONIC_ATTACK_PROBABILITY;
    };
}
#endif

//-----------------------------------------------------------------------------
//
//  demon.cpp
```

```
//

#include "demon.h"
#include <iostream>

using namespace std;

namespace cs_creature {

    const double demon::DEMONIC_ATTACK_PROBABILITY = 0.25;

    demon::demon(){
    }




    demon::demon(int newStrength, int newHitpoints)
    : creature(newStrength, newHitpoints){
    }




    string demon::getSpecies() const {
        return "demon";
    }




    int demon::getDamage() const {
        int damage = creature::getDamage();
        cout << " attacks for " << damage << " points!" << endl;
        if (rand() % 100 * 0.01 < DEMONIC_ATTACK_PROBABILITY) {
            damage = damage + DEMONIC_ATTACK_DAMAGE;
            cout << "Demonic attack inflicts "
                << DEMONIC_ATTACK_DAMAGE
                << " additional damage points!" << endl;
        }
        return damage;
    }
}
//--------------------------------------------------------------------------------------
//
//  cyberdemon.h
//

#ifndef cyberdemon_h
#define cyberdemon_h

#include "demon.h"
#include <string>

namespace cs_creature {

    class cyberdemon: public demon {
    public:
        cyberdemon();
        cyberdemon(int newStrength, int newHitpoints);
        int getDamage() const;
        std::string getSpecies() const;
    };
}
#endif

//--------------------------------------------------------------------------------------
//
//  cyberdemon.cpp
//
#include "cyberdemon.h"

#include <iostream>

using namespace std;

namespace cs_creature {

    cyberdemon::cyberdemon(){
    }




    cyberdemon::cyberdemon(int newStrength, int newHitpoints)
    : demon(newStrength, newHitpoints){
    }
```

```
    string cyberdemon::getSpecies() const {
        return "cyberdemon";
    }




    int cyberdemon::getDamage() const {
        cout << "The cyberdemon";

        int damage = demon::getDamage();
        return damage;
    }
}
//-----------------------------------------------------------------------------
//
//  balrog.h
//

#ifndef balrog_h
#define balrog_h

#include "demon.h"
#include <string>

namespace cs_creature {

    class balrog: public demon {
    public:
        balrog();
        balrog(int newStrength, int newHitpoints);
        int getDamage() const;
        std::string getSpecies() const;
    };
}
#endif

//-----------------------------------------------------------------------------
//
//  balrog.cpp
//
#include "balrog.h"
#include <iostream>
using namespace std;

namespace cs_creature {

    balrog::balrog(){
    }




    balrog::balrog(int newStrength, int newHitpoints)
    : demon(newStrength, newHitpoints){
    }




    string balrog::getSpecies() const {
        return "balrog";
    }




    int balrog::getDamage() const {

        cout << "The balrog";

        int damage = demon::getDamage();

        int damage2 = (rand() % getStrength()) + 1;
        cout << "Balrog speed attack inflicts " << damage2 << " additional damage points!" << endl;
        damage += damage2;
        return damage;
    }
}


//-----------------------------------------------------------------------------
//
```

```
// a8.cpp   client program
//
#include "human.h"
#include "elf.h"
#include "cyberdemon.h"
#include "balrog.h"
#include <ctime>
#include <iostream>

using namespace cs_creature;
using namespace std;

int main() {
    srand(time(0));

    human h1;
    elf e1;
    cyberdemon c1;
    balrog b1;

    human h(20, 30);
    elf e(40, 50);
    cyberdemon c(60, 70);
    balrog b(80, 90);


    cout << "default human strength/hitpoints: " << h1.getStrength() << "/" << h1.getHitpoints() << endl;
    cout << "default elf strength/hitpoints: " << e1.getStrength() << "/" << e1.getHitpoints() << endl;
    cout << "default cyberdemon strength/hitpoints: " << c1.getStrength() << "/" << c1.getHitpoints() << endl;
    cout << "default balrog strength/hitpoints: " << b1.getStrength() << "/" << b1.getHitpoints() << endl;
    cout << "non-default human strength/hitpoints: " << h.getStrength() << "/" << h.getHitpoints() << endl;
    cout << "non-default elf strength/hitpoints: " << e.getStrength() << "/" << e.getHitpoints() << endl;
    cout << "non-default cyberdemon strength/hitpoints: " << c.getStrength() << "/" << c.getHitpoints() << endl;
    cout << "non-default balrog strength/hitpoints: " << b.getStrength() << "/" << b.getHitpoints() << endl;
    cout << endl << endl;

    cout << "Examples of " << h.getSpecies() << " damage: " << endl;
    for (int i = 0; i < 10; i++){
        int damage = h.getDamage();
        cout << " Total damage = " << damage << endl;
        cout << endl;
    }
    cout << endl;



    cout << "Examples of " << e.getSpecies() << " damage: " << endl;
    for (int i = 0; i < 10; i++){
        int damage = e.getDamage();
        cout << " Total damage = " << damage << endl;
        cout << endl;
    }
    cout << endl;



    cout << "Examples of " << c.getSpecies() << " damage: " << endl;
    for (int i = 0; i < 10; i++){
        int damage = c.getDamage();
        cout << " Total damage = " << damage << endl;
        cout << endl;
    }
    cout << endl;



    cout << "Examples of " << b.getSpecies() << " damage: " << endl;
    for (int i = 0; i < 10; i++){
        int damage = b.getDamage();
        cout << " Total damage = " << damage << endl;
        cout << endl;
    }
    cout << endl;
}
```