```
#include <iostream>

#include <ctime>

#include "Creature.h"

#include "Wizard.h"

#include "Elf.h"

#include "Dwarf.h"

#include "Demon.h"

using namespace std;


int main() //main method.

{

   Creature* creatures[4]; //creature array

            int creature1, creature2; // the selection of the creature the player selects.

            int input;

            int player1_input, player2_input; //player 1 and 2 inputs.

            int winner; //the winner of the game.

            int round=0; //the number of rounds there are.

            int wizardMagicalPowerMissNextAttack=0; //the wizard can cause the opponent to miss
their next attack

            bool creatureLost[4] = { false };

            bool creatureWin[4] = { false };

            string name;

            //double strength, hitPoints, magicalDamage, armor, shield, height, speed; //creatures
attributes.

            double wizard_armor, elf_agility, dwarf_armor, demon_armor; //spcial ability of each
creature.

            int wandCheck; //checking if the wizard have a wand or not?

            string creature1_name, creature2_name, wandDetermine; //creature names.

            bool wand; //used for wand validation if the wizard have the wand or not?


            // DECLARE WIZARDS
```

```
	Wizard wizard1("Wizard1", 12, 100, 100, 10, wand, 1);
	//string name, double strength, double hitPoints, double health, double magicalDamage,
bool wand, double armour
	Wizard wizard2("Wizard2", 20, 200, 200, 20, wand, 5);



	// Used for Operator overloading due to the requirement
	Wizard wizard3=wizard1+wizard2;

	// Declare elfs
	Elf elf1("Elf1", 15, 100, 100, 3);
	//string name, double strength, double hitPoints, double health, double agility
	Elf elf2("Elf2", 20, 200, 200, 2);

	// Used for Operator overloading due to the requirement
	Elf elf3=elf1-elf2;

	// DECLARE DWARFS
	Dwarf dwarf1("Dwarf1", 20, 40, 10, 10, 10);
	//string name, double strength, double hitPoints, double health, double invisibility, double
armour
	Dwarf dwarf2("Dwarf2", 25, 3, 10, 2, 2);

	// Used for Operator overloading due to the requirement
	Dwarf dwarf3=dwarf1*dwarf2;

	// DECLARE DEMON
	Demon demon1("Demon1", 25, 140, 140, 10, 5);

	// POINT EACH OBJECT USING POLYMORPHISM
	creatures[0] = &wizard3; // Pointing to wizard
	creatures[1] = &elf3; // Pointing to elf
	creatures[2] = &dwarf3; // Pointing to dwarf
	creatures[3] = &demon1; // Pointing to a demon

	//Dynamic Cast
	Wizard* casted_wizard = dynamic_cast<Wizard*>(creatures[0]);
	//Elf* casted_elf = dynamic_cast<Elf*>(creatures[1]);
	//Dwarf* casted_dwarf = dynamic_cast<Dwarf*>(creatures[2]);
	//Demon* casted_demon = dynamic_cast<Demon*>(creatures[3]);

	// Selection of two creatures to fight.
	cout << "+----------------------------------------+" << endl;
	cout << "| Welcome to fight game between creatures |" << endl;
	cout << "+----------------------------------------+" << endl;
	cout << endl;

	cout << "+----------------------------------------+" << endl;
	cout << "|   Please, choose the FIRST creature.    |" << endl;
	cout << "|   1) Wizard                             |" << endl;
	cout << "|   2) Elf                                |" << endl;
	cout << "|   3) Dwarf                              |" << endl;
	cout << "|   4) Demon                              |" << endl;
	cout << "+----------------------------------------+" << endl;
	cout << "> ";
	cin >> creature1;

	// VALIDATION
	while (creatureLost[creature1-1] == true || creature1 < 1 || creature1 > 4) // FIRST
CREATURE VALIDATION
	{
		cout << "+----------------------------------------+" << endl;
		cout << "|            WRONG INPUT                  |" << endl;
```

```
			cout << "+----------------------------------------+" << endl;
			cout << "|   Please, choose the FIRST creature.    |" << endl;
			cout << "|   1) Wizard                          |" << endl;
			cout << "|   2) Elf                             |" << endl;
			cout << "|   3) Dwarf                           |" << endl;
			cout << "|   4) Demon                           |" << endl;
			cout << "+----------------------------------------+" << endl;
			cout << "> ";
			cin >> creature1;
		}

		cout << "+----------------------------------------+" << endl;
		cout << "|   Please, choose the SECOND creature.   |" << endl;
		cout << "|   1) Wizard                          |" << endl;
		cout << "|   2) Elf                             |" << endl;
		cout << "|   3) Dwarf                           |" << endl;
		cout << "|   4) Demon                           |" << endl;
		cout << "+----------------------------------------+" << endl;
		cout << "> ";
		cin >> creature2;

		// VALIDATION
		while (creatureLost[creature2 - 1] == true || creature2 < 1 || creature2 > 4 || creature1 ==
	creature2)
		{
			cout << "+----------------------------------------+" << endl;
			cout << "|            WRONG INPUT              |" << endl;
			cout << "+----------------------------------------+" << endl;
			cout << "|   Please, choose the SECOND creature.   |" << endl;
			cout << "|   1) Wizard                          |" << endl;
			cout << "|   2) Elf                             |" << endl;
```

```
            cout << "|   3) Dwarf                        |" << endl;
            cout << "|   4) Demon                        |" << endl;
            cout << "+----------------------------------------+" << endl;
            cout << "> ";
            cin >> creature2;
    }

    // FIGHT BETWEEN TWO CREATURES
    do
    {
            // KEEPING TRACK OF THE ROUNDS.
            round++;

            // G
            if (creature1 == 1)
            {
                    creature1_name = "Wizard";
                    wizard_armor = wizard3.getArmour();

                    do{
     cout << "Does the Wizard have a Wand? Enter '0' for No or '1' for Yes"<< endl;
     cin >> wandCheck;
     if(wandCheck == 0) {
        wand = false;
     } else if (wandCheck == 1) {
        wand = true;
     }
  } while(wandCheck < 0 || wandCheck >1);
            }

            else if (creature1 == 2)
```

```
{
        creature1_name = "Elf";
        elf_agility = elf3.getAgility();
}

else if (creature1 == 3)
{
        creature1_name = "Dwarf";
        dwarf_armor = dwarf3.getArmour();
}

else if (creature1 == 4)
{
        creature1_name = "Demon";
        demon_armor = demon1.getArmour();
}



if (creature2 == 1)
{
        creature2_name = "Wizard";
        wizard_armor = wizard3.getArmour();
}

else if (creature2 == 2)
{
        creature2_name = "Elf";
        elf_agility = elf3.getAgility();
}

else if (creature2 == 3)
```

```
		{
			creature2_name = "Dwarf";
			dwarf_armor = dwarf3.getArmour();
		}

		else if (creature2 == 4)
		{
			creature2_name = "Demon";
			demon_armor = demon1.getArmour();
		}

		//
		double creatureAbilitiesArr[4]={wizard_armor, elf_agility, dwarf_armor,
demon_armor};

		// Getting hit points from the creatures.
		double hp1 = creatures[creature1 - 1]->getHitPoints();
		double hp2 = creatures[creature2 - 1]->getHitPoints();

		// Battle menu
		while (hp1 > 0 && hp2 > 0)
		{
			cout << "+----------------------------------------+" << endl;
			cout << "|           FIGHT INFORMATION           |" << endl;
			cout << "+----------------------------------------+" << endl;
			cout << "PLAYER 1 : " << creature1_name << endl;
			cout << "PLAYER 2 : " << creature2_name << endl;

			// LET THE USER CHOOSE THE OPTION
			cout << "+----------------------------------------+" << endl;
			cout << "|    Please, choose PLAYER 1's option    |" << endl;
```

```
                        cout << "|         1) ATTACK   2) ESCAPE         |" << endl;
                        cout << "+---------------------------------------+" << endl;
                        cout << "> ";
                        cin >> player1_input;

                        if (player1_input == 1) // ATTACK
                        {
                                if(wizardMagicalPowerMissNextAttack==1)
                                {
                                        cout << "+-----------------------------------------------------------------
-----------+" << endl;
                                        cout << "| BECAUSE OF WIZARD MAGICAL POWER, THIS
TURN, YOU CANNOT ATTACK THE OPPONENT  |" << endl;
                                        cout << "+-----------------------------------------------------------------
-----------+" << endl;
                                        wizardMagicalPowerMissNextAttack=0;
                                }

                                else
                                {
                                        // GENERATE THE RANDOM NUMBER BETWEEN 1 TO 100
FOR PROBABILITY
                                        srand(time(0));
                                        int randomNumber=(rand()%100)+1;
                                        cout << "+---------------------------------------+" << endl;
                                        cout << "|     Player 1 attacks Player 2 !!!       |" << endl;
                                        cout << "+---------------------------------------+" << endl;

                                        if(creature1==1) // IF THE CREATURE IS WIZARD
                                        {
                                                // IF THE WIZARD DOES NOT HAVE THE WAND, 50%
chance
                                                if(casted_wizard->getWand()==false)
                                        {

            if(randomNumber>0&&randomNumber<=50)
                                                {
                                                        hp2 = creatures[creature2 - 1]-
>getHitPoints();
                                                        hp2 -= casted_wizard-
>getMagicalDamage();
                                                        hp2 +=
creatureAbilitiesArr[creature2 - 1];

            wizardMagicalPowerMissNextAttack=1;

            wizardMagicalPower(*casted_wizard); // calling the friend function
                                                }

                                                else
                                                {
                                                        hp2 = creatures[creature2 - 1]-
>getHitPoints();
                                                        hp2 -= creatures[creature1 - 1]-
>getStrength();
                                                        hp2 +=
creatureAbilitiesArr[creature2 - 1];
                                                }
                                        }

                                        // if the wizard have a wand.
                                        else
                                        {

            if(randomNumber>0&&randomNumber<=90)
                                                {
                                                        hp2 = creatures[creature2 - 1]-
>getHitPoints();
```

```
                                                                        hp2 -= casted_wizard-
>getMagicalDamage();
                                                                        hp2 +=
creatureAbilitiesArr[creature2 - 1];

        wizardMagicalPowerMissNextAttack=1;

        wizardMagicalPower(*casted_wizard); // calling the friend function
                                                                }

                                                                else
                                                                {
                                                                        hp2 = creatures[creature2 - 1]-
>getHitPoints();
                                                                        hp2 -= creatures[creature1 - 1]-
>getStrength();
                                                                        hp2 +=
creatureAbilitiesArr[creature2 - 1];
                                                                }
                                                        }
                                                }

                                                else if(creature1==2) // elf
                                                {
                                                        if(randomNumber>0&&randomNumber<=10)
                                                        {
                                                                hp2 = creatures[creature2 - 1]-
>getHitPoints();
                                                                hp2 -= creatures[creature1 - 1]-
>getStrength();
                                                                hp2 -= creatures[creature1 - 1]-
>getStrength();
                                                                hp2 += creatureAbilitiesArr[creature2 - 1];
                                                                cout << "+----------------------------------------+"
<< endl;
```

```
                        cout << "|   ELF ARE VERY FAST TO ATTACK
TWICE!!   |" << endl;
                        cout << "+----------------------------------------+"
<< endl;
                    }

                    else
                    {
                        hp2 = creatures[creature2 - 1]-
>getHitPoints();
                        hp2 -= creatures[creature1 - 1]-
>getStrength();
                        hp2 += creatureAbilitiesArr[creature2 - 1];
                    }
                }

                else if(creature1==3) // dwarf
                {
                    if(randomNumber>0&&randomNumber<=10)
                    {
                        hp2 = creatures[creature2 - 1]-
>getHitPoints();
                        hp2 -= (creatures[creature1 - 1]-
>getStrength()*2);
                        hp2 += creatureAbilitiesArr[creature2 - 1];
                        cout << "+----------------------------------------+"
<< endl;
                        cout << "|      DWARF INFLICTS DOUBLE
DAMAGE!!     |" << endl;
                        cout << "+----------------------------------------+"
<< endl;
                    }

                    else
```

```
                        {
                                hp2 = creatures[creature2 - 1]-
>getHitPoints();
                                hp2 -= creatures[creature1 - 1]-
>getStrength();
                                hp2 += creatureAbilitiesArr[creature2 - 1];
                        }
                }
                else if(creature1==4) // demon
                {
                        if(randomNumber>0&&randomNumber<=5)
                        {
                                hp2 = creatures[creature2 - 1]-
>getHitPoints();
                                hp2 -= (creatures[creature1 - 1]-
>getStrength()+50);
                                hp2 += creatureAbilitiesArr[creature2 - 1];
                                cout << "+----------------------------------------+"
<< endl;
                                cout << "|  DEMONS INFLICTS 50
ADDTIONATL DAMAGE!  |" << endl;
                                cout << "+----------------------------------------+"
<< endl;
                        }

                        else
                        {
                                hp2 = creatures[creature2 - 1]-
>getHitPoints();
                                hp2 -= creatures[creature1 - 1]-
>getStrength();
                                hp2 += creatureAbilitiesArr[creature2 - 1];
                        }
                }
```

```
        }

        if(hp2<0) hp2=0;
        creatures[creature2 - 1]->setHitPoints(hp2);

        cout << "+----------------------------------------+" << endl;
        cout << "|          Player 1 Information          |" << endl;
        cout << "+----------------------------------------+" << endl;
        creatures[creature1 - 1]->display();

        cout << "+----------------------------------------+" << endl;
        cout << "|          Player 2 Information          |" << endl;
        cout << "+----------------------------------------+" << endl;
        creatures[creature2 - 1]->display();
}
else if (player1_input == 2) // ESCAPE
{
        cout << "Player 1 escapes!" << endl;
        break;
}
else // validation
{
        cout << "Your input is wrong!" << endl;
}

// validation for checking if the player 2 health is zero or not.
if (hp2 <= 0)
{
        cout << "Player 1 wins!" << endl;
        cout << "Player 2 is defeated!" << endl;
```

```
                creatureLost[creature2 - 1] = true;
                creatureWin[creature1 - 1] = true;
                winner=1;

                if (creature2 == 1) void *wizard3=NULL;
                else if (creature2 == 2) void *elf3=NULL;
                else if (creature2 == 3) void *dwarf3=NULL;
                else if (creature2 == 4) void *demon1=NULL;
                break;
            }

            // Player 2 options.
            cout << "+----------------------------------------+" << endl;
            cout << "|    Please, choose PLAYER 2's option     |" << endl;
            cout << "|         1) ATTACK   2) ESCAPE          |" << endl;
            cout << "+----------------------------------------+" << endl;
            cout << "> ";
            cin >> player2_input;

            if (player2_input == 1) // ATTACK
            {
                if(wizardMagicalPowerMissNextAttack==1)
                {
                    cout << "+----------------------------------------------------------------
-----------+" << endl;
                    cout << "| BECAUSE OF WIZARD MAGICAL POWER, THIS
TURN, YOU CANNOT ATTACK THE OPPONENT  |" << endl;
                    cout << "+----------------------------------------------------------------
-----------+" << endl;
                    wizardMagicalPowerMissNextAttack=0;
                }
            else
            {
                    // generation random number for the probability
                    srand(time(0));
                    int randomNumber=(rand()%100)+1;
                    cout << "+-----------------------------------------+" << endl;
                    cout << "|     Player 2 attacks Player 1 !!!       |" << endl;
                    cout << "+-----------------------------------------+" << endl;

                    if(creature2==1)
                    {
                            // Wizard doesn't have a wand, 50% chance
                            if(casted_wizard->getWand()==false)
                            {

    if(randomNumber>0&&randomNumber<=50)
                                    {
                                            hp1 = creatures[creature1 - 1]-
>getHitPoints();
                                            hp1 -= casted_wizard-
>getMagicalDamage();
                                            hp1 +=
creatureAbilitiesArr[creature1 - 1];

    wizardMagicalPowerMissNextAttack=1;
                                            cout << "+----------------------------------
-------+" << endl;
                                            cout << "|   WIZARD MAGICAL
DAMAGE WAS INFLICTED!  |" << endl;
                                            cout << "+----------------------------------
-------+" << endl;
                                    }

                                    else
```

```
                                                {
                                                        hp1 = creatures[creature1 - 1]-
>getHitPoints();
                                                        hp1 -= creatures[creature2 - 1]-
>getStrength();
                                                        hp1 +=
creatureAbilitiesArr[creature1 - 1];
                                                }
                                        }

                                        // if the wizard have a wand the opponents attack
will be missed for 90%
                                        else
                                        {
        if(randomNumber>0&&randomNumber<=90)
                                                {
                                                        hp1 = creatures[creature1 - 1]-
>getHitPoints();
                                                        hp1 -= casted_wizard-
>getMagicalDamage();
                                                        hp1 +=
creatureAbilitiesArr[creature1 - 1];

        wizardMagicalPowerMissNextAttack=1;
                                                        cout << "+----------------------------------
-------+" << endl;
                                                        cout << "|   WIZARD MAGICAL
DAMAGE WAS INFLICTED!  |" << endl;
                                                        cout << "+----------------------------------
-------+" << endl;
                                                }

                                                else
                                                {
```

```
                                                                hp1 = creatures[creature1 - 1]-
>getHitPoints();
                                                                hp1 -= creatures[creature2 - 1]-
>getStrength();
                                                                hp1 +=
creatureAbilitiesArr[creature1 - 1];
                                                        }
                                                }
                                        }
                                        else if(creature2==2) // elf
                                        {
                                                if(randomNumber>0&&randomNumber<=10)
                                                {
                                                        hp1 = creatures[creature1 - 1]-
>getHitPoints();
                                                        hp1 -= creatures[creature2 - 1]-
>getStrength();
                                                        hp1 -= creatures[creature2 - 1]-
>getStrength();
                                                        hp1 += creatureAbilitiesArr[creature1 - 1];
                                                        cout << "+----------------------------------------+"
<< endl;
                                                        cout << "|   ELF ARE VERY FAST TO ATTACK
TWICE!!   |" << endl;
                                                        cout << "+----------------------------------------+"
<< endl;
                                                }

                                                else
                                                {
                                                        hp1 = creatures[creature1 - 1]-
>getHitPoints();
                                                        hp1 -= creatures[creature2 - 1]-
>getStrength();
                                                        hp1 += creatureAbilitiesArr[creature1 - 1];
```

```
					}
				}
				else if(creature2==3) // dwarf.
				{
					if(randomNumber>0&&randomNumber<=10)
					{
						hp1 = creatures[creature1 - 1]-
>getHitPoints();
						hp1 -= creatures[creature2 - 1]-
>getStrength();
						hp1 += creatureAbilitiesArr[creature1 - 1];
						cout << "+----------------------------------------+"
<< endl;
						cout << "|     DWARF INFLICTS DOUBLE
DAMAGE!!     |" << endl;
						cout << "+----------------------------------------+"
<< endl;
					}

					else
					{
						hp1 = creatures[creature1 - 1]-
>getHitPoints();
						hp1 -= creatures[creature2 - 1]-
>getStrength();
						hp1 += creatureAbilitiesArr[creature1 - 1];
					}
				}
				else if(creature2==4) // demon
				{
					if(randomNumber>0&&randomNumber<=5)
					{
						hp1 = creatures[creature1 - 1]-
>getHitPoints();
```

```
                                        hp1 -= (creatures[creature2 - 1]-
>getStrength()+50);
                                        hp1 += creatureAbilitiesArr[creature1 - 1];
                                        cout << "+----------------------------------------+"
<< endl;
                                        cout << "|  DEMONS INFLICTS 50
ADDITIONAL DAMAGE!  |" << endl;
                                        cout << "+----------------------------------------+"
<< endl;
                                }

                                else
                                {
                                        hp1 = creatures[creature1 - 1]-
>getHitPoints();
                                        hp1 -= creatures[creature2 - 1]-
>getStrength();
                                        hp1 += creatureAbilitiesArr[creature1 - 1];
                                }
                        }
                }

                if(hp1<0) hp1=0;
                creatures[creature1 - 1]->setHitPoints(hp1);

                cout << "+----------------------------------------+" << endl;
                cout << "|          Player 1 Information          |" << endl;
                cout << "+----------------------------------------+" << endl;
                creatures[creature1 - 1]->display();

                cout << "+----------------------------------------+" << endl;
                cout << "|          Player 2 Information          |" << endl;
                cout << "+----------------------------------------+" << endl;
```

```
        creatures[creature2 - 1]->display();
}

else if (player2_input == 2)
{
        cout << "Player 2 escapes!" << endl;
        break;
}

else
{
        cout << "Your input is wrong!" << endl;
}

// to check if the player's hp is below 0 or not.
if (hp1 <= 0)
{
        cout << endl;
        cout << "Player 2 wins!" << endl;
        cout << "Player 1 is defeated!" << endl;

        creatureLost[creature1 - 1] = true;
        creatureWin[creature2 - 1] = true;
        winner=2;

        // Making objects of creatures which are killed to NULL.
        if (creature1 == 1) void *wizard3=NULL;
        else if (creature1 == 2) void *elf3=NULL;
        else if (creature1 == 3) void *dwarf3=NULL;
        else if (creature1 == 4) void *demon1=NULL;
        break;
```

```
			}
		}

		// ASsking users if they want to continue or not?
		if(round<3)
		{
			cout << "\nDo you want to continue? Enter '1' for YES OR '0' for NO : ";
			cin >> input;

			if(input==1)
			{
				if(winner==1)
				{
				   cout << "\nThe "<< creature1_name << " will fight the next
opponent.\n" << endl;
					cout << "+-------------------------------------------------------+" <<
endl;
					cout << "|   Please, select a NEW opponent for the winner to
fight. |" << endl;
					cout << "|   1) Wizard                                |" << endl;
					cout << "|   2) Elf                                   |" << endl;
					cout << "|   3) Dwarf                                 |" << endl;
					cout << "|   4) Demon                                 |" << endl;
					cout << "+-------------------------------------------------------+" <<
endl;
					cout << "> ";
					cin >> creature2;

					while (creatureLost[creature2 - 1] == true || creature2 < 1
|| creature2 > 4 || creature1 == creature2) ///creature 1 has been changed.
					{
						cout << "+---------------------------------------+" << endl;
```

```
                                        cout << "|   Your input is wrong!!!              |" <<
endl;
                                        cout << "+----------------------------------------+" << endl;
                                        cout << "|   Please, choose the second creature.   |"
<< endl;
                                        cout << "|   1) Wizard                        |" << endl;
                                        cout << "|   2) Elf                        |" << endl;
                                        cout << "|   3) Dwarf                        |" << endl;
                                        cout << "|   4) Demon                        |" << endl;
                                        cout << "+----------------------------------------+" << endl;
                                        cout << "> ";
                                        cin >> creature2;
                                }
                        }

                        else if(winner==2)
                        {
                                cout << "The "<< creature2_name << " will fight the next
opponent." << endl;
                                cout << "+---------------------------------------------------------+" <<
endl;
                                cout << "|   Please, select a NEW opponent for the winner to
fight. |" << endl;
                                cout << "|   1) Wizard                                |" << endl;
                                cout << "|   2) Elf                                |" << endl;
                                cout << "|   3) Dwarf                                |" << endl;
                                cout << "|   4) Demon                                |" << endl;
                                cout << "+---------------------------------------------------------+" <<
endl;
                                cout << "> ";
                                cin >> creature1;
```

```
                                        while (creatureLost[creature1 - 1] == true || creature1 < 1
|| creature1 > 4 || creature1 == creature2)
                                        {
                                                cout << "+----------------------------------------+" << endl;
                                                cout << "|   Your input is wrong!!!               |" <<
endl;
                                                cout << "+----------------------------------------+" << endl;
                                                cout << "|   Please, choose the second creature.   |"
<< endl;
                                                cout << "|   1) Wizard                          |" << endl;
                                                cout << "|   2) Elf                             |" << endl;
                                                cout << "|   3) Dwarf                           |" << endl;
                                                cout << "|   4) Demon                           |" << endl;
                                                cout << "+----------------------------------------+" << endl;
                                                cout << "> ";
                                                cin >> creature1;
                                        }
                                }
                        }


                        else if(input==0)
                        {
                                cout << "End the game." << endl;
                        }
                }


                else
                {
                        cout << "A total round of games are completed! Thank you very much!" <<
endl;
                }
        } while (input != 0 && round!=3);
```

```
            return 0;
}

//Name-Tasfique
//Student ID-5886429
///CREATURE CLASS.
#include <iostream>
#include "Creature.h"

using namespace std;

Creature::Creature() // Default constructor
{
            name = "";
            strength = 0.0;
            hitPoints = 0.0;
            health = 0.0;
}

Creature::Creature(string name, double strength, double hitPoints, double health) : name(name),
strength(strength), hitPoints(hitPoints), health(health) {
}

void Creature::setName(string name) {
   this->name = name;
}

void Creature::setStrength(double strength) {
   this->strength = strength;
}
```

```
void Creature::setHitPoints(double hitPoints) {
   this->hitPoints = hitPoints;
}

void Creature::setHealth(double health) {
   this->health = health;
}

string Creature::getName() {
   return name;
}

double Creature::getStrength() {
   return strength;
}

double Creature::getHitPoints() {
   return hitPoints;
}

double Creature::getHealth() {
   return health;
}

Creature::~Creature() {}
void Creature::display() {}


//Name-Tasfique
//Student ID-5886429
```

```
#ifndef CREATURE_H

#define CREATURE_H

#include <iostream>

#include <ctime>


using namespace std;


class Creature {


private:

   string name;

   double strength;

   double hitPoints;

   double health;


public:

   Creature(); // Default constructor initialization



   Creature(string, double, double, double); // non-default constructor.

   ///setters

   void setName(string);

   void setStrength(double);

   void setHitPoints(double);

   void setHealth(double);

   ///getters

   string getName();

   double getStrength();

   double getHealth();

   double getHitPoints();

   virtual ~Creature(); // virtual destructor

   virtual void display();
};


#endif // CREATURE_H

//Name-Tasfique
//Student ID-5886429
#include <iostream>
#include "Wizard.h"

using namespace std;

Wizard::Wizard() : Creature() {
   magicalDamage = 0.0;
   wand = false;
   armour = 0.0;
}

Wizard::Wizard(string name, double strength, double hitPoints, double health, double
magicalDamage, bool wand, double armour) : Creature(name, strength, hitPoints, health),
wand(wand), magicalDamage(magicalDamage), armour(armour) {
} //Wizard wizard1("Wizard1", 12, 100, 100, 10, true, 1);

void Wizard::setMagicalDamage(double magicalDamage) {
   this->magicalDamage = magicalDamage;
}

void Wizard::setWand(bool wand) {
   this->wand = wand;
}
```

```
void Wizard::setArmour(double armour) {
   this->armour = armour;
}

double Wizard::getMagicalDamage()
{
        return magicalDamage;
}

bool Wizard::getWand()
{
        return wand;
}

double Wizard::getArmour()
{
        return armour;
}

Wizard Wizard::operator+(Wizard const &w)
{
        return Wizard(getName(), getStrength(), getHitPoints(), getHealth(),
magicalDamage+w.magicalDamage, wand, armour+w.armour);
}

void Wizard::display()
{
        cout << "Name of the Wizard : " << getName() << endl;
        cout << "Strength : " << getStrength() << endl;
        cout << "Hit Points : " << getHitPoints() << "/" << getHealth() << endl;
        cout << "Magical Damage : " << getMagicalDamage() << endl;
        if(wand) cout << "Possesion of Wand : TRUE " << endl;
        else cout << "POSSESSION OF WAND : FALSE " << endl;
        cout << "Armour : " << getArmour() << endl;
}

void wizardMagicalPower(Wizard const &w)
{
        cout << "+----------------------------------------+" << endl;
        cout << "|   WIZARD MAGICAL DAMAGE WAS INFLICTED!  |" << endl;
        cout << "+----------------------------------------+" << endl;
        cout << "Wizard Magical Damage : " << w.magicalDamage << endl;
}
//Name-Tasfique
//Student ID-5886429
#ifndef WIZARD_H
#define WIZARD_H
#include <iostream>
#include <ctime>
#include "Creature.h"

class Wizard : public Creature
{
private:
   double magicalDamage;
   bool wand;
   double armour;

public:
   Wizard();
   Wizard(string, double, double, double, double, bool, double);
   ~Wizard() {};
   Wizard operator+(const Wizard &w);
   void setMagicalDamage(double);
   void setWand(bool);
   void setArmour(double);
   double getMagicalDamage();
   bool getWand();
   double getArmour();
   void display() override;
   friend void wizardMagicalPower(const Wizard &w);
};


#endif // CREATURE_H
//Name-Tasfique
//Student ID-5886429
#include <iostream>
#include "Dwarf.h"

Dwarf::Dwarf() : Creature()
{
	invisibility = 0.0;
	armour = 0.0;
}

Dwarf::Dwarf(string name, double strength, double hitPoints, double health, double invisibility,
double armour) : Creature(name, strength, hitPoints, health), invisibility(invisibility), armour(armour)
{}

void Dwarf::setInvisibility(double invisibility)
{
	this->invisibility = invisibility;
}
```

```
void Dwarf::setArmour(double armour)

{

	this->armour = armour;

}


double Dwarf::getInvisibility()

{

	return invisibility;

}


double Dwarf::getArmour()

{

	return armour;

}


Dwarf Dwarf::operator*(Dwarf const &e) //arithmetic operator overloading.

{

	return Dwarf(getName(), getStrength(), getHitPoints(), getHealth(), invisibility*e.invisibility, 
armour*e.armour);

}


void Dwarf::display() //display

{

	cout << "Name : " << getName() << endl;

	cout << "Strength : " << getStrength() << endl;

	cout << "Hit Points : " << getHitPoints() << "/" << getHealth() << endl;

	cout << "Invisibility : " << invisibility << endl;

	cout << "Armour : " << armour << endl;

}

//Name-Tasfique
```

```
//Student ID-5886429
#ifndef DWARF_H
#define DWARF_H
#include "Creature.h"

class Dwarf : public Creature {

private:
   //double n ;
   double armour;
   double invisibility;

public:
   Dwarf();
   Dwarf(string, double, double, double, double, double);
   ~Dwarf() {}; //destructor.
   Dwarf operator*(const Dwarf& d);
   //setter
   void setInvisibility(double);
   void setArmour(double);
   //getter
   double getInvisibility();
   double getArmour();
   //display
   void display() override;
};


#endif
//Name-Tasfique
//Student ID-5886429
```

```
#include "Elf.h"


Elf::Elf(string name, double strength, double hitPoints, double health, double agility) :
Creature(name, strength, hitPoints, health), agility(agility) {}


void Elf::setAgility(double agility)
{
	this->agility = agility;
}


double Elf::getAgility()
{
	return agility;
}


Elf Elf::operator-(Elf const &e)
{
	return Elf(getName(), getStrength(), getHitPoints(), getHealth(), agility-e.agility);
}


void Elf::display() //display method.
{
	cout << "NAME : " << getName() << endl;
	cout << "STRENGTH : " << getStrength() << endl;
	cout << "HIT POINTS : " << getHitPoints() << "/" << getHealth() << endl;
	cout << "ARMOR : " << agility << endl;
}


//Name-Tasfique
//Student ID-5886429
```

```
#ifndef ELF_H

#define ELF_H

#include "Creature.h"


class Elf : public Creature {


private:

   double agility;

public:

   Elf();

   Elf(string, double, double, double, double); //constructor.

   ~Elf() {}; //destructor.

   Elf operator-(const Elf& e); //operator overloading

   void setAgility(double);

   double getAgility();

   void display() override;


};


#endif

//Name-Tasfique

//Student ID-5886429

#include <iostream>

#include "Demon.h"


Demon::Demon() : Creature()

{

        speed = 0.0;

        armour = 0.0;

}
```

```
Demon::Demon(string name, double strength, double hitPoints, double health, double speed,
double armour) : Creature(name, strength, hitPoints, health), speed(speed), armour(armour) {}


void Demon::setSpeed(double speed)

{

	this->speed = speed;

}


void Demon::setArmour(double armour)

{

	this->armour = armour;

}


double Demon::getSpeed()

{

	return speed;

}


double Demon::getArmour()

{

	return armour;

}


void Demon::display()

{

	cout << "Name : " << getName() << endl;

	cout << "Strength : " << getStrength() << endl;

	cout << "Hit Point : " << getHitPoints() << "/" << getHealth() << endl;

	cout << "Speed : " << speed << endl;

	cout << "Armour : " << armour << endl;

}
```

```
//Name-Tasfique

//Student ID-5886429

#ifndef DEMON_H

#define DEMON_H

#include "Creature.h"


class Demon : public Creature

{

        private:

                double speed;

                double armour;


        public:

                Demon();

                Demon(string, double, double, double, double, double);

                ~Demon() {};

                void setSpeed(double);

                void setArmour(double);

                double getSpeed();

                double getArmour();

                void display() override;

};


#endif
```