Github Link: <https://github.com/MrChang2/CS2560-Spring2019>

Source Code:

#pragma once

#include <string>

class Human

{

friend class Parent;

friend class Child;

private:

std::string name;

int age;

char sex;

Human();

protected:

Human(std::string n, int a, char c);

virtual ~Human();

public:

void setName(std::string n);

std::string getName();

void setAge(int a);

int getAge();

void setSex(char c);

char getSex();

};

Human::Human()

{

}

Human::Human(std::string n, int a, char c) {

name = n;

age = a;

sex = c;

}

Human::~Human()

{

std::cout << "Destructing Human" << std::endl;

}

void Human::setName(std::string n) {

name = n;

}

std::string Human::getName() {

return name;

}

void Human::setAge(int a) {

age = a;

}

int Human::getAge() {

return age;

}

void Human::setSex(char c) {

sex = c;

}

char Human::getSex() {

return sex;

}

#pragma once

#include "Human.h"

#include <string>

#include <iostream>

class Child : public Human

{

friend class Parent;

private:

std::string momName;

std::string dadName;

int allowance;

Child();

public:

Child(std::string n, int a, char c, std::string mn, std::string dn);

~Child();

int getAllowance() const;

void printParent();

};

#include "Child.h"

Child::Child(std::string n, int a, char c, std::string mn, std::string dn) {

this->setName(n);

this->setAge(a);

this->setSex(c);

allowance = 0;

momName = mn;

dadName = dn;

}

Child::~Child()

{

}

int Child::getAllowance() const {

return allowance;

}

void Child::printParent() {

std::cout << "Mom's name: " << momName << std::endl;

std::cout << "Dad's name: " << dadName << std::endl;

}

#pragma once

#include "Human.h"

#include "Child.h"

#include <vector>

#include <string>

#include <iostream>

class Parent : public Human

{

private:

std::vector<Child> children;

Parent();

public:

Parent(std::string n, int a, char c);

~Parent();

void printChild();

void setChild(Child c);

void setAllowance(int a, Child c);

};

#include "Parent.h"

Parent::Parent(std::string n, int a, char c) {

this->setName(n);

this->setAge(a);

this->setSex(c);

}

Parent::~Parent()

{

}

void Parent::printChild() {

for (int x = 0; x < children.size(); x++) {

std::cout << "Child " << x + 1 << ": " << children.at(x).getName() << std::endl;

}

}

void Parent::setChild(Child c) {

children.push\_back(c);

}

void Parent::setAllowance(int a, Child c) {

for (int x = 0; x < children.capacity(); x++) {

if (children[x].getName() == c.getName()) {

std::cout << "Changing allowance of " << children[x].getName() << " to " << a << std::endl;

children.at(x).allowance = a;

break;

}

}

}

#include <iostream>

#include "Child.h"

#include "Parent.h"

//void printInfo(Human h) {

//std::cout << "Name: " << h.getName() << std::endl;

//std::cout << "Age: " << h.getAge() << std::endl;

//std::cout << "Sex: " << h.getSex() << std::endl;

//}

int main() {

Parent Homer("Homer", 36, 'M');

Parent Marge("Marge", 34, 'F');

Child Lisa("Lisa", 12, 'F', "Marge", "Homer");

Child Bart("Bart", 10, 'M', "Marge", "Homer");

Child Maggie("Maggie", 3, 'F', "Marge", "Homer");

Homer.setChild(Lisa);

Homer.setChild(Bart);

Homer.setChild(Maggie);

Marge.setChild(Lisa);

Marge.setChild(Bart);

Marge.setChild(Maggie);

std::cout << "Bart's allowance is " << Bart.getAllowance() << std::endl;

Homer.setAllowance(5, Bart);

std::cout << "Bart's new allowance is " << Bart.getAllowance() << std::endl;

std::cout << "Bart's Parents: " << std::endl;

Bart.printParent();

//printInfo(Marge);

//printInfo(Homer);

char k;

std::cout << "Enter any key to quit." << std::endl;

std::cin >> k;

return 0;

}

Output:

