**Student Header**

#ifndef STUDENT\_H

#define STUDENT\_H

#include <string>

using namespace std;

class Student

{

public:

Student();

Student(string name, int age, int units);

string getName();

void setName(string name);

int getAge();

void setAge(int age);

int getUnits();

void setUnits(int units);

double tuition(double feePerUnit);

void printStudent();

private:

string name;

int age;

int units;

};

#endif

**Student Implementation**

#include "Student.h"

#include <iostream>

using namespace std;

Student::Student()

{

name = "John";

age = 20;

units = 3;

}

Student::Student(string name, int age, int units)

{

this->name = name;

this->age = age;

this->units = units;

}

string Student::getName()

{

return name;

}

void Student::setName(string name)

{

this->name = name;

}

int Student::getAge()

{

return age;

}

void Student::setAge(int age)

{

this->age = age;

}

int Student::getUnits()

{

return units;

}

void Student::setUnits(int units)

{

this->units = units;

}

double Student::tuition(double feePerUnit)

{

return feePerUnit \* units;

}

void Student::printStudent()

{

cout << name << " is " << age << " years old.\n" << name

<< "'s tuition cost is $" << tuition(100) << " for "

<< getUnits() << " units.\n";

}

**Course Header**

#ifndef COURSE\_H

#define COURSE\_H

#include <string>

#include "Student.h"

using namespace std;

class Course

{

public:

Course();

Course(string name, Student enrolled);

void setName(string name);

string getName();

void setEnrolled(Student enrolled);

Student getEnrolled();

void printCourse();

private:

string name;

Student enrolled;

};

#endif

**Course Implementation**

#include "Course.h"

#include "Student.h"

#include <string>

#include <iostream>

using namespace std;

Course::Course()

{

name = "class";

enrolled = Student();

}

Course::Course(string name, Student enrolled)

{

this->name = name;

this->enrolled = enrolled;

}

void Course::setName(string name)

{

this->name = name;

}

string Course::getName()

{

return name;

}

void Course::setEnrolled(Student enrolled)

{

this->enrolled = enrolled;

}

Student Course::getEnrolled()

{

return enrolled;

}

void Course::printCourse()

{

enrolled.printStudent();

cout << enrolled.getName() << " is a student of course " << name << ".\n";

}

**Main**

#include <iostream>

#include "Student.h"

#include "Course.h"

using namespace std;

// function for Question 4 comparing two student objects

// and returning the student object with lower amount of units

Student smaller(Student stu1, Student stu2)

{

if (stu1.getUnits() < stu2.getUnits())

return stu1;

else

return stu2;

}

int main()

{

// Question 1. printing a student object with name = Tim, age = 20 and units = 5

cout << "Question 1\n";

Student s1("Tim", 20, 5);

s1.printStudent();

cout << endl;

// Question 2. creating a pointer to the student object

// part a. pointing the created student object

cout << "Question 2\n";

Student\* sptr = &s1;

// part b. calling the tuition method

cout << sptr->getName() << "'s tuition cost is now increased to "

<< sptr->tuition(200) << ".\n\n";

// part c. dynamically allocating a new student

Student\* s2 = new Student("Mark", 30, 8);

sptr = s2;

// part d. calling the print method and deleting the pointer

sptr->printStudent();

delete sptr;

cout << endl;

//Question 3. creating a course object with one student and print

cout << "Question 3\n";

Student s3("Jerry", 25, 4);

Course c1("CSC 252", s3);

c1.printCourse();

cout << endl;

// Question 4. comparing two students and returning the one with smaller number of units

cout << "Question 4\n";

cout << "Between " << s1.getName() << " and " << s3.getName() << ", "

<< smaller(s1, s3).getName() << " has lower amount of units.\n";

cout << endl;

// Question 5. dynamically allocating an array of 5 integers

cout << "Question 5\n";

int\* iptr = new int[5];

// prompting the user to enter five integers

cout << "Enter five integers: ";

for (int i = 0; i < 5; i++)

cin >> \*(iptr + i);

// displaying the values through loop

for (int i = 0; i < 5; i++)

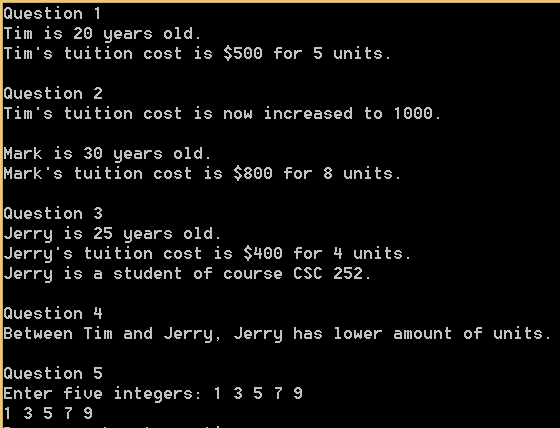
cout << iptr[i] << " ";

cout << endl;

system("pause");

return 0;

}

****