**QUESTION 1**

#include <iostream>

#include <string>

#include <sstream>

#include <ctime>

using namespace std;

class Person {

public:

Person(std::string name, int year) : name(name), yearOfBirth(year), ss(std::ios\_base::out) {}

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

int getYearOfBirth() const { return yearOfBirth; }

virtual std::string toString() const {

ss.str("");

ss << "Name: " << name << ", Year of Birth: " << yearOfBirth;

return ss.str();

}

bool equals(const Person &p) const {

return name == p.name && yearOfBirth == p.yearOfBirth;

}

private:

std::string name;

int yearOfBirth;

mutable std::stringstream ss;

};

class Student : public Person {

public:

Student(std::string name, int year, std::string major)

: Person(name, year), major(major) {}

std::string getMajor() const { return major; }

std::string toString() const override {

return Person::toString() + ", Major: " + major;

}

bool equals(const Student &s) const {

return Person::equals(s) && major == s.major;

}

private:

std::string major;

};

class Instructor : public Person {

public:

Instructor(std::string name, int year, int salary)

: Person(name, year), salary(salary) {}

int getSalary() const { return salary; }

std::string toString() const override {

return Person::toString() + ", Salary: ";// + std::to\_string(salary);

}

bool equals(const Instructor &i) const {

return Person::equals(i) && salary == i.salary;

}

private:

int salary;

};

int main() {

Person p("John Doe", 1990);

Student s("Jane Doe", 1995, "Computer Science");

Instructor i("Bob Smith", 1980, 100000);

std::cout << p.toString() << std::endl;

std::cout << s.toString() << std::endl;

std::cout << i.toString() << std::endl;

std::cout << "p and s are equal: " << (p.equals(s) ? "True" : "False") << std::endl;

std::cout << "s and s are equal: " << (s.equals(s) ? "True" : "False") << std::endl;

std::cout << "i and i are equal: " << (i.equals(i) ? "True" : "False") << std::endl;

return 0;

}

**QUESTION 2**

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

class GeometricObject {

// class definition for GeometricObject

};

class Triangle : public GeometricObject {

public:

std::string color;

bool filled;

void setColor(string newValue) {

color = newValue;

}

void setFill(bool newValue) {

filled = newValue;

}

Triangle() : side1(1.0), side2(1.0), side3(1.0) {}

Triangle(double side1, double side2, double side3) : side1(side1), side2(side2), side3(side3) {}

double getSide1() const { return side1; }

double getSide2() const { return side2; }

double getSide3() const { return side3; }

void setSide1(double side1) { this->side1 = side1; }

void setSide2(double side2) { this->side2 = side2; }

void setSide3(double side3) { this->side3 = side3; }

double getPerimeter() const { return side1 + side2 + side3; }

std::string toString() const {

std::stringstream ss;

ss << "Triangle: side1 = " << side1 << " side2 = " << side2 << " side3 = " << side3;

return ss.str();

}

private:

double side1;

double side2;

double side3;

};

int main() {

double side1, side2, side3;

std::string color;

bool filled;

int filled\_entry;

Triangle triangle(side1, side2, side3);

std::cout<<"What is the color ? \n";

std::cin>>color;

std::cout<<"Is the Triangle filled ?, 1 for yes and 0 for No \n";

std::cin>>filled\_entry;

if(filled\_entry == 1){

filled = true;

}else if(filled\_entry==0){

filled = false;

}

//MUTATOR(MODIFY) FUNCTIONS OF THE PROGRAM

triangle.setColor(color);

triangle.setFill(filled);

std::cout << "Perimeter of triangle: " << triangle.getPerimeter() << std::endl;

std::cout << triangle.toString() << std::endl;

return 0;

}