Question no 1:

#include <iostream>

// Base class

class Shape {

public:

virtual double calculateArea() {

return 0;

}

};

// Derived class Circle

class Circle : public Shape {

private:

double radius;

public:

Circle(double r) : radius(r) {}

double calculateArea() override {

return 3.14159 \* radius \* radius;

}

};

// Derived class Rectangle

class Rectangle : public Shape {

private:

double length;

double width;

public:

Rectangle(double l, double w) : length(l), width(w) {}

double calculateArea() override {

return length \* width;

}

};

int main() {

Shape \*shape1 = new Circle(5);

Shape \*shape2 = new Rectangle(4, 6);

std::cout << "Area of Circle: " << shape1->calculateArea() << std::endl;

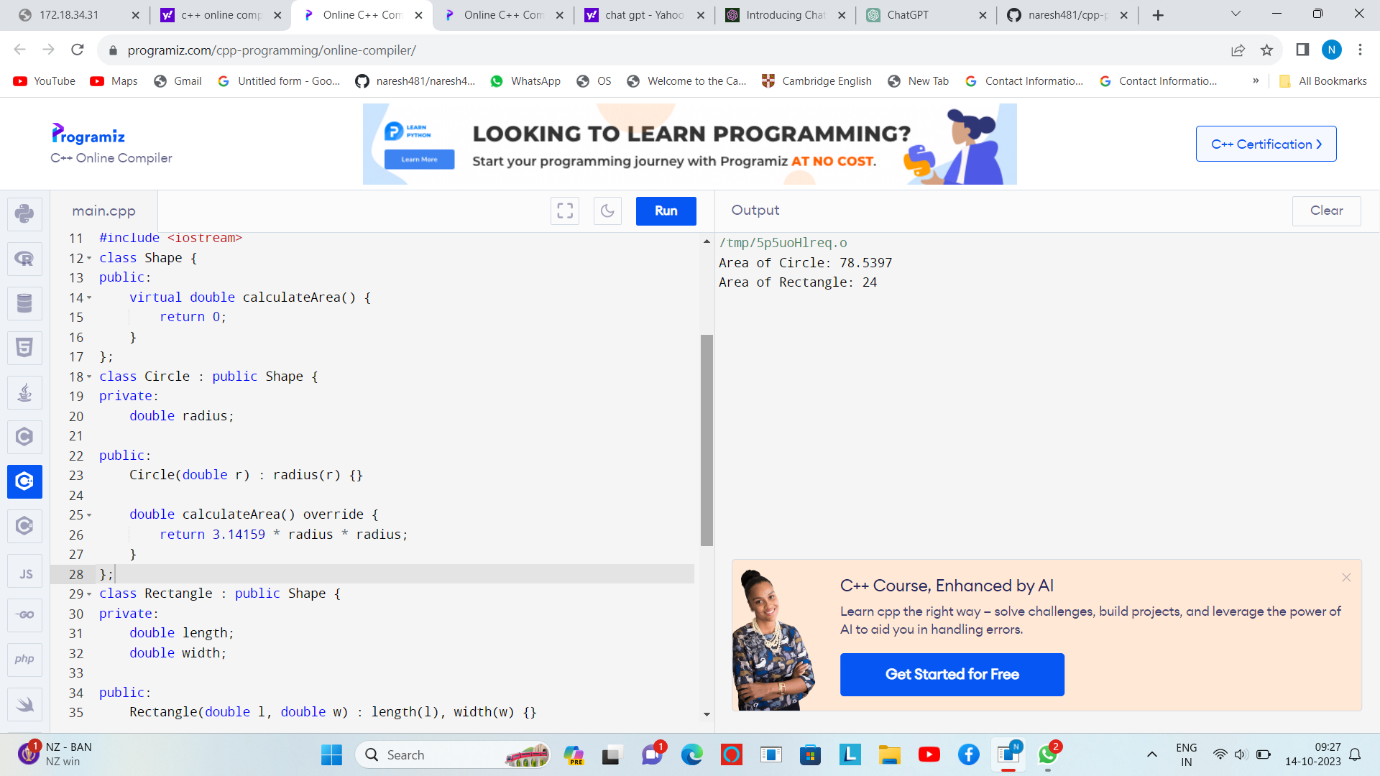
std::cout << "Area of Rectangle: " << shape2->calculateArea() << std::endl;

delete shape1;

delete shape2;

return 0;

}



Question no 2:

#include <iostream>

class MyClass;

void displayValue(const MyClass& obj);

class MyClass {

private:

int value;

public:

MyClass(int val) : value(val) {}

friend void displayValue(const MyClass& obj);

};

void displayValue(const MyClass& obj) {

std::cout << "The value in MyClass is: " << obj.value << std::endl;

}

int main() {

MyClass obj(5);

displayValue(obj);

return 0;

}

