Program 1:

#include <iostream>

#include <cmath>

class Shape {

public:

virtual float area() = 0;

};

class Circle : public Shape {

private:

float radius;

public:

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

float area() override {

return 3.14159 \* radius \* radius;

}

};

class Rectangle : public Shape {

private:

float length, width;

public:

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

float area() override {

return length \* width;

}

};

class Cylinder : public Rectangle {

private:

float height;

public:

Cylinder(float r, float h) : Rectangle(r, r), height(h) {}

float volume() {

return area() \* height;

}

};

int main() {

Circle circle(5);

Rectangle rectangle(4, 6);

Cylinder cylinder(3, 7);

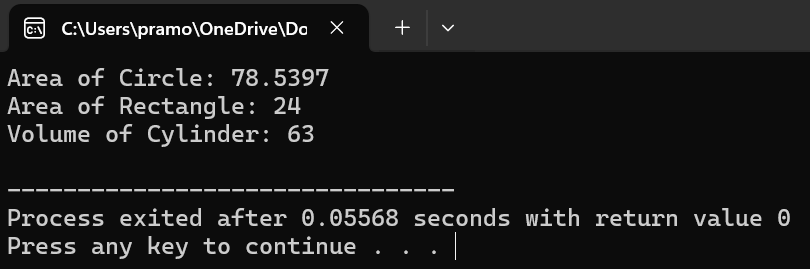
std::cout << "Area of Circle: " << circle.area() << std::endl;

std::cout << "Area of Rectangle: " << rectangle.area() << std::endl;

std::cout << "Volume of Cylinder: " << cylinder.volume() << std::endl;

return 0;

}



Program 2:

#include <iostream>

#include <string>

using namespace std;

class Employee {

private:

string name;

int employeeID;

float salary;

public:

Employee(string n, int id, float s) : name(n), employeeID(id), salary(s) {}

void setDetails(string n, int id, float s) {

name = n;

employeeID = id;

salary = s;

}

void displayDetails() {

cout << "Name: " << name << endl;

cout << "Employee ID: " << employeeID << endl;

cout << "Salary: $" << salary << endl;

}

};

int main() {

const int numEmployees = 3;

Employee employees[numEmployees] = {

Employee("John", 1001, 50000),

Employee("Alice", 1002, 60000),

Employee("Bob", 1003, 55000)

};

for (int i = 0; i < numEmployees; ++i) {

cout << "Employee " << i + 1 << " details:" << endl;

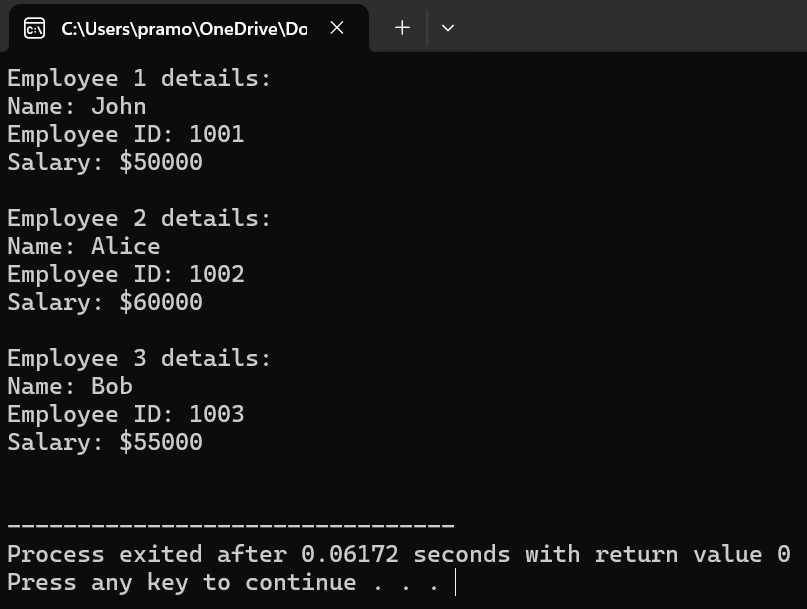
employees[i].displayDetails();

cout << endl;

}

return 0;

}



Program 3:

#include <iostream>

class Number {

protected:

int num;

public:

Number(int n) : num(n) {}

virtual int square() {

return num \* num;

}

};

class Square : public Number {

public:

Square(int n) : Number(n) {}

int square() override {

return num \* num;

}

};

class Cube : public Number {

public:

Cube(int n) : Number(n) {}

int cube() {

return num \* num \* num;

}

};

int main() {

int number = 5;

Square square(number);

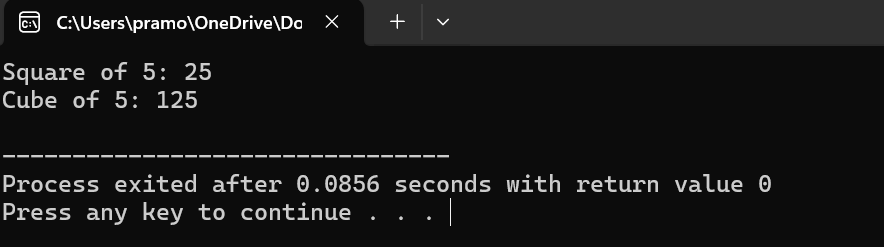
Cube cube(number);

std::cout << "Square of " << number << ": " << square.square() << std::endl;

std::cout << "Cube of " << number << ": " << cube.cube() << std::endl;

return 0;

}



Program 4:

#include <iostream>

int\* findGreatest(int\* num1, int\* num2, int\* num3) {

if (\*num1 >= \*num2 && \*num1 >= \*num3)

return num1;

else if (\*num2 >= \*num1 && \*num2 >= \*num3)

return num2;

else

return num3;

}

int main() {

int num1, num2, num3;

std::cout << "Enter three numbers: ";

std::cin >> num1 >> num2 >> num3;

int\* greatest = findGreatest(&num1, &num2, &num3);

std::cout << "The greatest number is: " << \*greatest << std::endl;

return 0;

}

