Program 1:

#include <iostream>

class AddAmount {

protected:

double amount;

public:

AddAmount() : amount(50.0) {}

double getAmount() const {

return amount;

}

};

class DerivedAddAmount : public AddAmount {

public:

DerivedAddAmount(double additionalAmount) {

amount += additionalAmount;

}

};

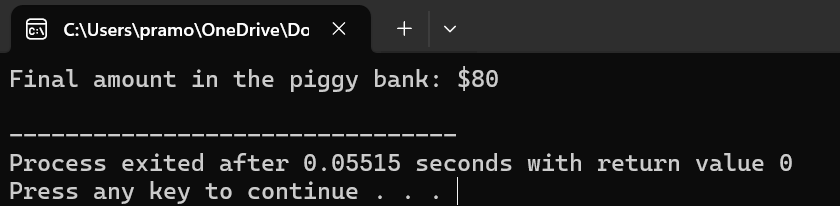
int main() {

DerivedAddAmount piggyBank(30.0);

std::cout << "Final amount in the piggy bank: $" << piggyBank.getAmount() << std::endl;

return 0;

}



Program 2:

#include <iostream>

#include <string>

class Person {

protected:

std::string name;

int age;

public:

void getData(const std::string& n, int a) {

name = n;

age = a;

}

virtual void displayData() const {

std::cout << "Name: " << name << ", Age: " << age << std::endl;

}

virtual float calculateBonus() const = 0;

};

class Admin : public Person {

protected:

float basicSalary;

public:

void getData(const std::string& n, int a, float salary) {

Person::getData(n, a);

basicSalary = salary;

}

void displayData() const override {

Person::displayData();

std::cout << "Basic Salary: $" << basicSalary << std::endl;

}

float calculateBonus() const override {

return basicSalary \* 0.1;

}

};

class Account : public Person {

protected:

float monthlyIncome;

public:

void getData(const std::string& n, int a, float income) {

Person::getData(n, a);

monthlyIncome = income;

}

void displayData() const override {

Person::displayData();

std::cout << "Monthly Income: $" << monthlyIncome << std::endl;

}

float calculateBonus() const override {

return monthlyIncome \* 0.05;

}

};

class Master : public Admin, public Account {

public:

void getData(const std::string& n, int a, float salary, float income) {

Admin::getData(n, a, salary);

Account::getData(n, a, income);

}

void displayData() const override {

std::cout << "Details of Master Employee:" << std::endl;

Admin::displayData();

Account::displayData();

}

float calculateBonus() const override {

return Admin::calculateBonus() + Account::calculateBonus();

}

};

int main() {

Master employee;

employee.getData("John Doe", 35, 5000.0, 3000.0);

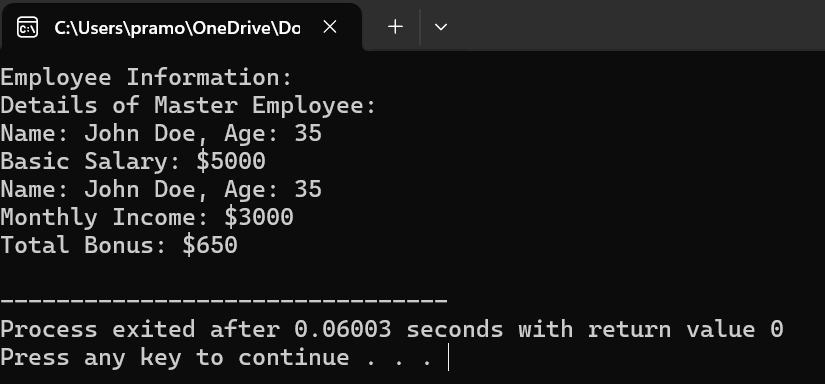
std::cout << "Employee Information:" << std::endl;

employee.displayData();

std::cout << "Total Bonus: $" << employee.calculateBonus() << std::endl;

return 0;

}



Program 3:

#include <iostream>

#include <string>

class Employee {

protected:

std::string name;

int emp\_id;

float basicSalary;

public:

Employee(const std::string& n, int id, float salary)

: name(n), emp\_id(id), basicSalary(salary) {}

virtual void calculatePay(float& grossPay, float& netPay) const = 0;

};

class SalaryCalculator : public Employee {

public:

SalaryCalculator(const std::string& n, int id, float salary)

: Employee(n, id, salary) {}

void calculatePay(float& grossPay, float& netPay) const override {

grossPay = basicSalary + (basicSalary \* 0.1);

netPay = grossPay - (grossPay \* 0.2);

}

};

int main() {

std::string name;

int emp\_id;

float basicSalary;

std::cout << "Enter Employee Name: ";

std::getline(std::cin, name);

std::cout << "Enter Employee ID: ";

std::cin >> emp\_id;

std::cout << "Enter Basic Salary: ";

std::cin >> basicSalary;

SalaryCalculator employee(name, emp\_id, basicSalary);

float grossPay, netPay;

employee.calculatePay(grossPay, netPay);

std::cout << "\nEmployee Details:" << std::endl;

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

std::cout << "Employee ID: " << emp\_id << std::endl;

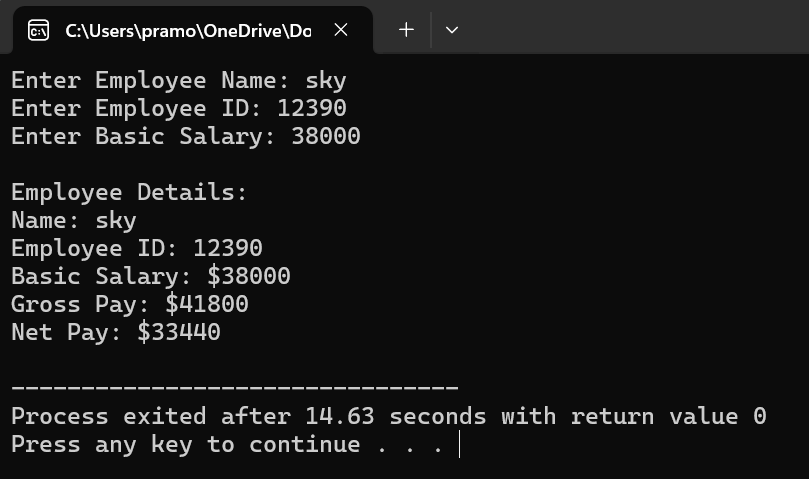
std::cout << "Basic Salary: $" << basicSalary << std::endl;

std::cout << "Gross Pay: $" << grossPay << std::endl;

std::cout << "Net Pay: $" << netPay << std::endl;

return 0;

}



Program 4:

#include <iostream>

class Shape {

public:

virtual float area() const = 0;

virtual float volume() const = 0;

};

class Rectangle : public Shape {

protected:

float length;

float width;

public:

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

float area() const override {

return length \* width;

}

float volume() const override {

return 0;

}

};

class Cuboid : public Rectangle {

protected:

float height;

public:

Cuboid(float l, float w, float h) : Rectangle(l, w), height(h) {}

float area() const override {

return 2 \* (length \* width + width \* height + height \* length);

}

float volume() const override {

return length \* width \* height;

}

};

int main() {

float length, width, height;

std::cout << "Enter length of cuboid: ";

std::cin >> length;

std::cout << "Enter width of cuboid: ";

std::cin >> width;

std::cout << "Enter height of cuboid: ";

std::cin >> height;

Cuboid cuboid(length, width, height);

std::cout << "\nCuboid Details:" << std::endl;

std::cout << "Area: " << cuboid.area() << std::endl;

std::cout << "Volume: " << cuboid.volume() << std::endl;

return 0;

}

