МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

УНІВЕРСИТЕТ БАНКІВСЬКОЇ СПРАВИ

Лабораторна робота №5

з дисципліни : Об’єктно-орієнтоване програмування

Виконала студентка

1 курсу (скороченого терміну навчання)

Сорока Олена Олександрівна

КИЇВ 2020

**Lab 5.1.1 Classes and Objects in C++**

**Code:**

#include <iostream>

#include<cstdlib>

#include<string>

#include <algorithm>

using namespace std;

class Person

{

private:

string name;

int age;

public:

Person(string name, int age)

{

this->name = name;

this->age = age;

}

void print()

{

cout << "Meet " << name << "and his age is " << age << endl;

}

};

int main()

{

string name, repeat;

int age;

cout << "Enter person`s name: ";

getline(cin, name);

cout << "His age is? - ";

cin >> age;

Person person(name, age);

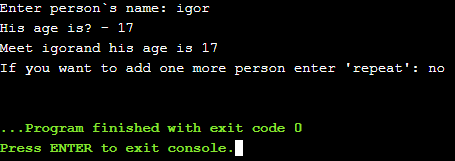
person.print();

cout << "If you want to add one more person enter 'repeat': ";

cin >> repeat;

return 0;

}



**Lab 5.1.2 Restricting access to object data**

**Code:**

#include <iostream>

#include<cstdlib>

#include<string>

#include <algorithm>

using namespace std;

class Square

{

private:

double side;

double area;

public:

void set\_data(double\* side)

{

this->side = \*side;

area = this->side \* this->side;

}

void print()

{

cout << "Square: side = " << side << " area = " << area << endl;

}

};

int main()

{

double side = 7.0;

Square a;

a.set\_data(&side);

a.print();

side = -74.0;

a.set\_data(&side);

a.print();

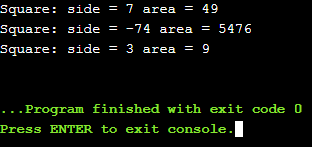
side = 3.0;

a.set\_data(&side);

a.print();

return 0;

}

****

**Code:**

#include <iostream>

#include<string>

#include<cmath>

using namespace std;

class AdHocSquare

{

private:

double side;

public:

AdHocSquare(double side)

{

this->side = side;

}

double get\_area()

{

return (this->side \* this->side);

}

void print()

{

cout << "Side - " << this->side << "\t" << "Area - " << get\_area() << endl;

}

};

class LazySquare

{

public:

LazySquare(double side, double prev\_side)

{

this->side = side;

this->area = pow(this->side, 2);

if (prev\_side == this->side)

this->side\_changed = false;

else

this->side\_changed = true;

}

void print()

{

if (!side\_changed)

cout << "We don`t need to do it, you have it higher" << endl;

else

cout << "Side - " << this->side << "\t" << "Area - " << this->area << endl;

}

private:

double side;

double area;

bool side\_changed;

};

int main()

{

double side, side1;

cout << "Enter square`s side: ";

cin >> side;

AdHocSquare square(side);

square.print();

cout << "Enter one more side: ";

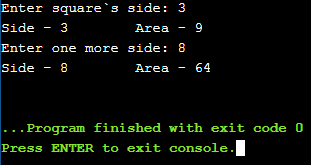
cin >> side1;

LazySquare square1(side1, side );

square1.print();

return 0;

}

****

**Lab 5.1.4 Classes and objects: ShopItemOrder**

**Code:**

#include <iostream>

#include<string>

#include<cmath>

using namespace std;

class ShopItemOrder

{

private:

string name;

float price, total\_price;

int amount;

public:

ShopItemOrder(string name, float price, int amount)

{

this->name = name;

this->price = price;

this->amount = amount;

this->total\_price = price \* amount;

}

void print()

{

cout << "You had bought " << this->amount << " of " << this->name << ". One unit of it costs " << this->price << " USD. You had spent " << this->total\_price << " USD." << endl;

}

};

int main()

{

string name;

float price;

int amount;

cout << "Enter the name: ";

cin.ignore();

getline(cin, name);

cout << "Enter the price of it: ";

cin >> price;

cout << "Enter the number of items, you want to buy: ";

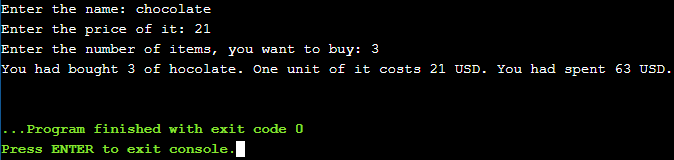
cin >> amount;

ShopItemOrder order(name, price, amount);

order.print();

return 0;

}

****