



**Al-Mustaqbal University**  
**College of Engineering and Technologies**  
**Prosthetics and Orthotics Engineering**



# **Computer Programming Laboratory**

## **Lec.1**

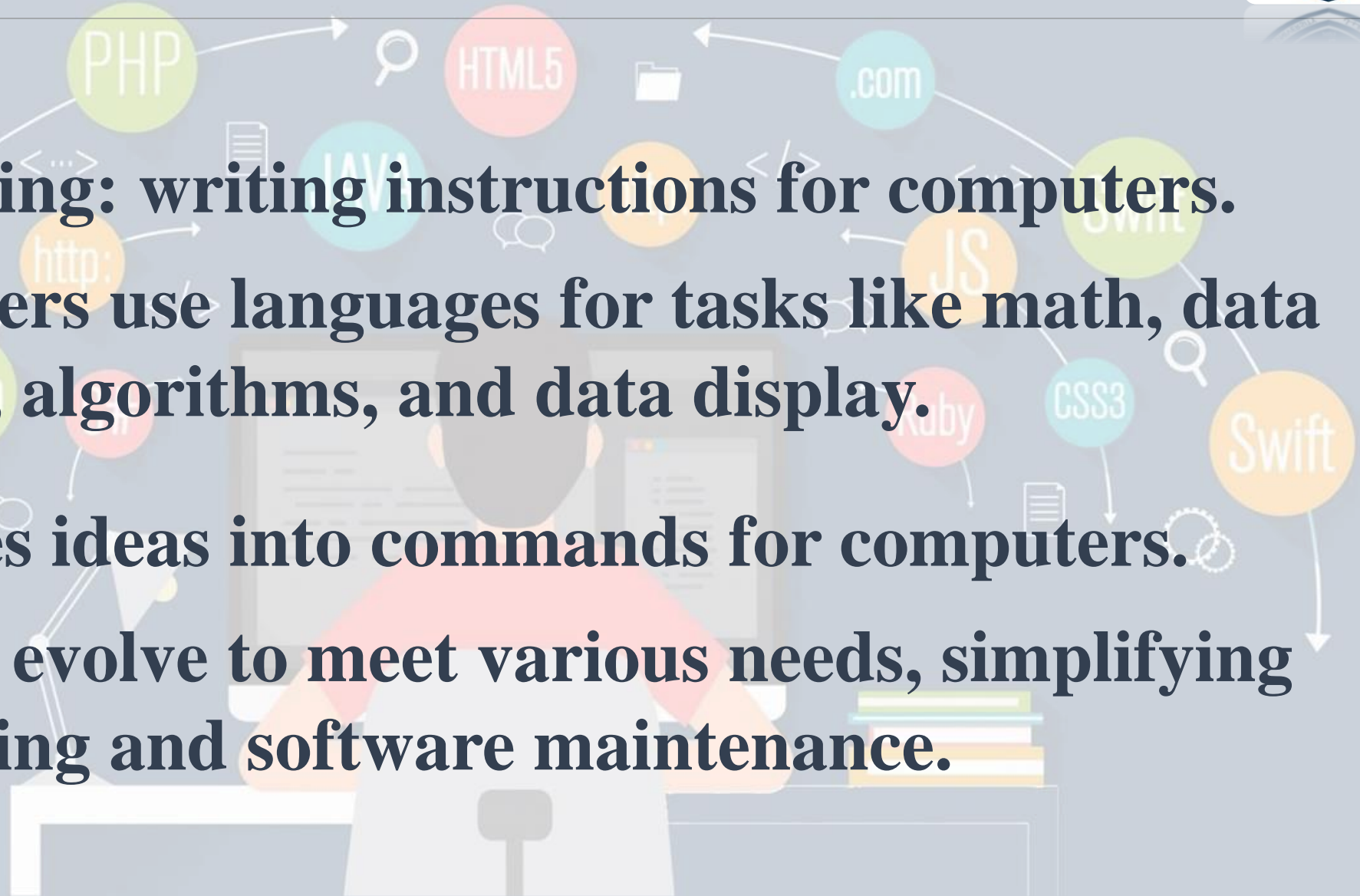
**Msc. Alyaa Abdul Kadhum**  
**Eng. Noor Adnan Madlool**



# Programming



- **Programming: writing instructions for computers.**
- **Programmers use languages for tasks like math, data processing, algorithms, and data display.**
- **It translates ideas into commands for computers.**
- **Languages evolve to meet various needs, simplifying programming and software maintenance.**





# C++



1. C++ is a powerful and comprehensive programming language used to develop computer applications, games, programs, and complex systems.
2. It is considered an evolution of the C language, and provides additional advantages such as object-oriented programming and direct memory manipulation, making it suitable for a wide range of applications.



# Visual Studio



1. Visual Studio: Microsoft's IDE tailored for C++ and other Microsoft languages.
2. The abbreviation IDE stands for Integrated Development Environment, which is a software environment that provides integrated tools for developing, editing, testing, and debugging programs.
3. Tools: Offers a comprehensive set including a code editor, debugger, compiler, and linker for C++ development.
4. Understanding: Crucial to interpret C++ effectively, requires grasp of syntax and structure.



# Computer Programming



## Understanding C++ code requires knowledge of:

1. Data types: Integers, floats, chars, and strings.
2. Variables: Declared with keywords like int, float, char, and string.
3. Expressions: Combinations of variables, operators (+, -, \*, /), comparison (==, !=, <, >, <=, >=), and logical (&&, ||, !).
4. Statements: Instructions for the computer, including assignments, control flow (if, for loops), functions, and returns.
5. Functions: Reusable blocks of code defined with the function keyword.
6. Classes and Objects: Support for object-oriented programming (OOP) with classes as blueprints and objects as instances encapsulating data and behavior.

Effective interpretation involves understanding these elements and their interactions within the code.



# Data Types



## FLOAT

رقم فيه علامة عشرية  
موجب او سالب

## CHARACTER

وده بنخزن فيه حرف واحد فقط  
بين علامتين تنصيص

## ARRAY

ودي للقائمة المرتبة

## INTEGER

رقم صحيح  
موجب زيرو سالب

## BOOLEAN

قيمة صحيحة او خاطئة  
TRUE FALSE

## STRING

وده بيتخزن فيه مجموعة من



# Data Types



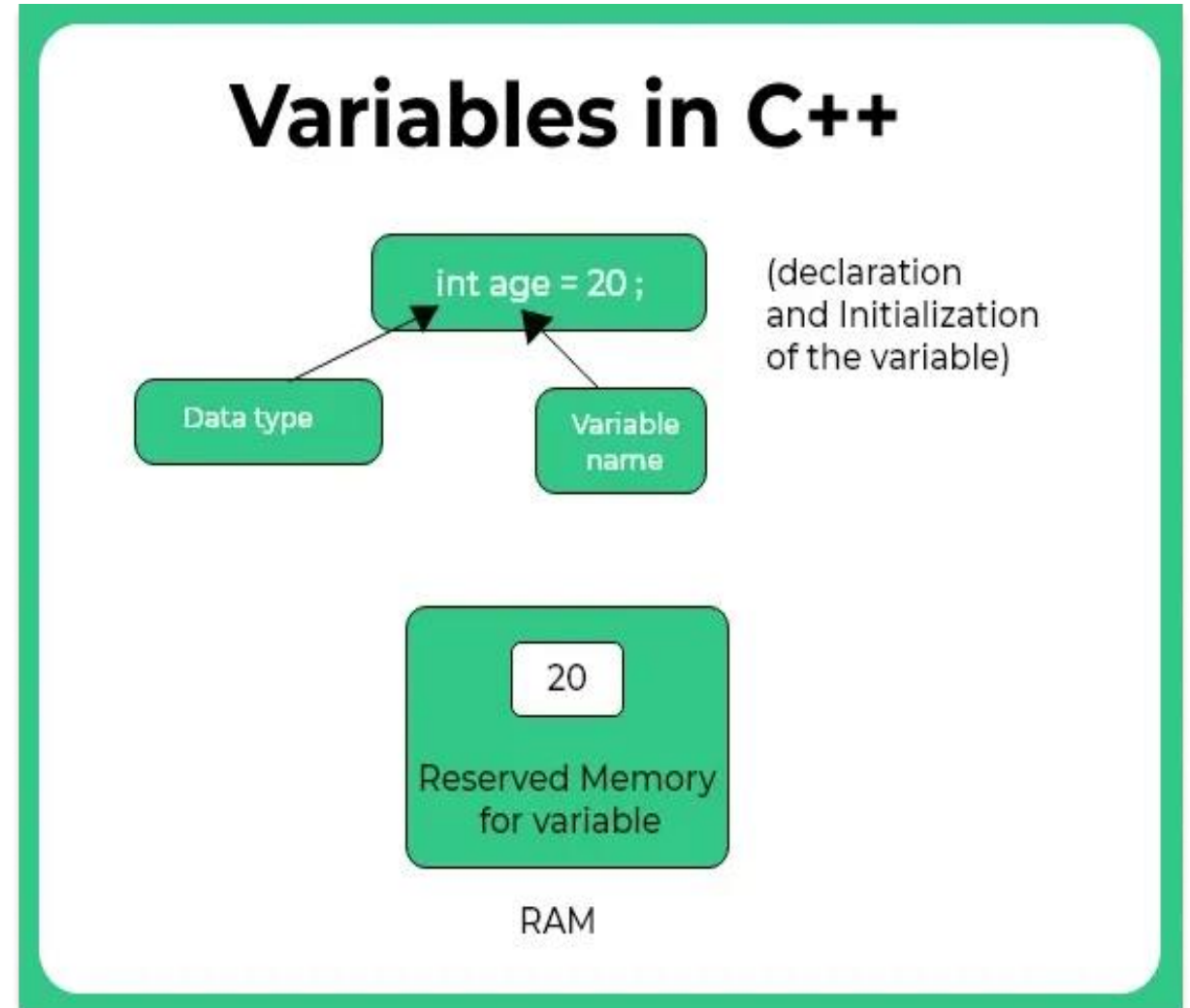
Type	Definition	Control Character	Limits
<b>int</b>	Integer		-2147483648 to 2147483647
<b>short</b>	Short Integer		-32768 to 32767
<b>long</b>	Long Integer	l or L	-2147483648 to 2147483647
<b>float</b>	Floating Decimal Number	f or F	1.17549e-038 to 3.40282e+038
<b>double</b>	Double Decimal Number		2.22507e-308 to 1.79769e+308
<b>long double</b>	Long Decimal Number		2.22507e-308 to 1.79769e+308
<b>char</b>	Character		-128 to 127
<b>unsigned int</b>	Unsigned Integer		0 to 4294967295
<b>unsigned short</b>	Unsigned Short Integer		0 to 65535
<b>unsigned long</b>	Unsigned Long Integer		0 to 4294967295
<b>unsigned char</b>	Unsigned Character		0 to 255
<b>bool</b>	True or False		True = 1 and False = 0



# Variables



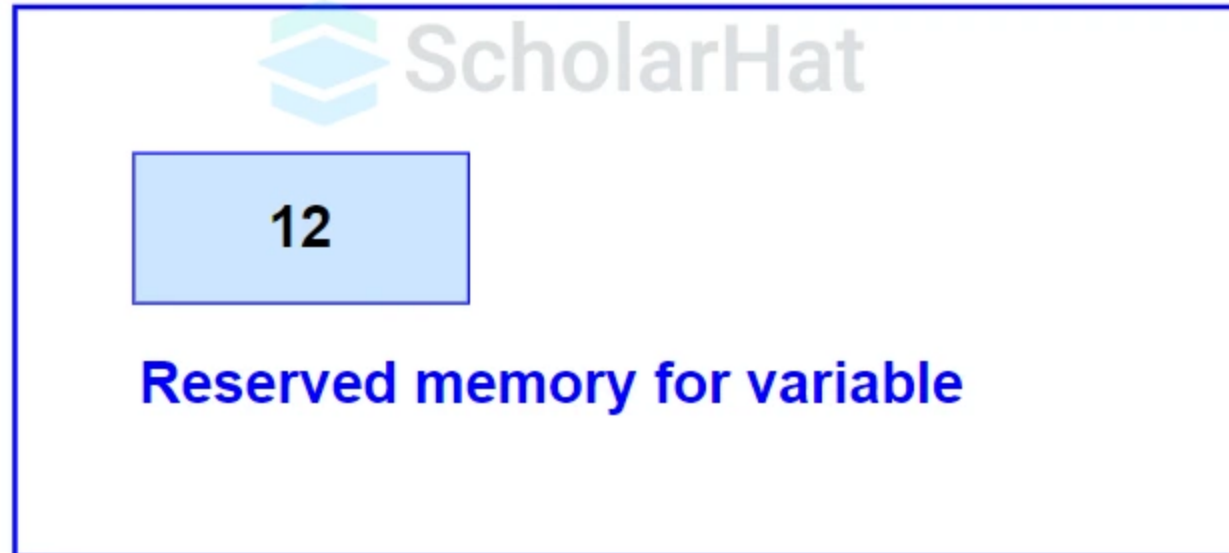
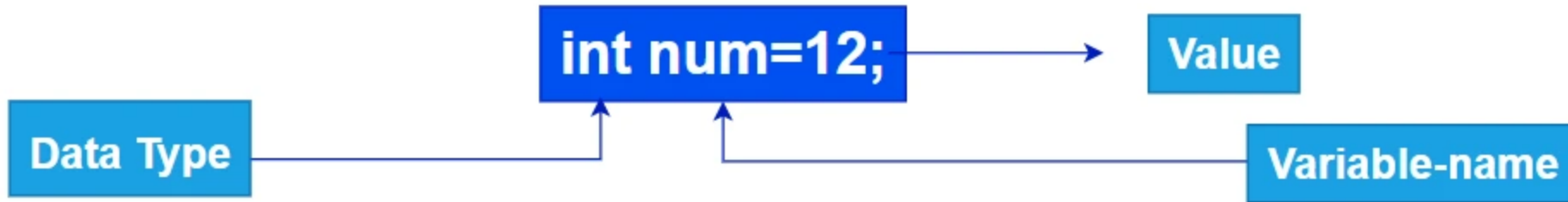
- It is a storage space that has a distinctive name into which we place a specific type of data
- Variables are containers for storing data values.
- In C++, there are different types of variables (defined with different keywords),







# Variables



**RAM**



# Computer Programming



## Sample program

```
#include <iostream>
using namespace std;

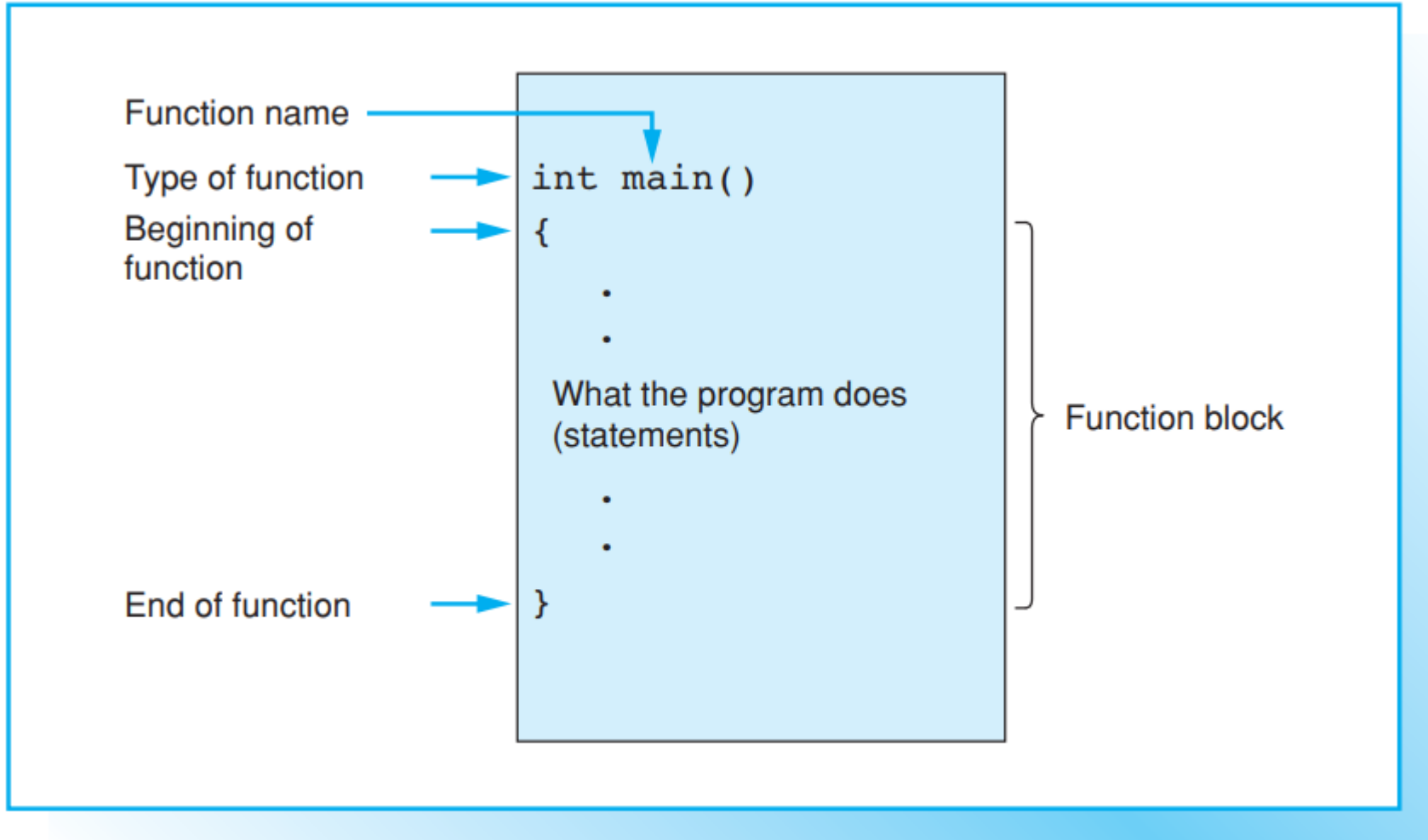
int main()
{
    cout << "Enjoy yourself with C++!" << endl;
    return 0;
}
```

## Screen output

Enjoy yourself with C++!



# Computer Programming





# Computer Programming



```
/*
 * A program with some functions and comments
 */

#include <iostream>
using namespace std;

void line(), message();           // Prototypes

int main()
{
    cout << "Hello! The program starts in main()."
          << endl;
    line();
    message();
    line();
    cout << "At the end of main()." << endl;

    return 0;
}

void line()                       // To draw a line.
{
    cout << "-----" << endl;
}

void message()                   // To display a message.
{
    cout << "In function message()." << endl;
}
```

## Screen output

```
Hello! The program starts in main().
-----
In function message().
-----
At the end of main().
```



# Computer Programming



## Program listing of exercise 3

```
#include <iostream>
using namespace std;

void pause();          // Prototype

int main()
{
    cout << endl << "Dear reader, "
         << endl << "have a ";
    pause();
    cout << "!" << endl;

    return 0;
}

void pause()
{
    cout << "BREAK";
}
```



# CODE



## Hello World C++ Program code

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```



# CODE



## Area and circumference of a circle and a rectangle c++ visual studio code

```
#include <iostream>
#include <cmath>

using namespace std;

const double PI = 3.14159;

int main() {
    // Circle
    double radius;
    cout << "Enter the radius of the circle: ";
    cin >> radius;

    double circle_area = PI * pow(radius, 2);
    double circle_circumference = 2 * PI * radius;

    cout << "Area of the circle: " << circle_area << endl;
    cout << "Circumference of the circle: " <<
    circle_circumference << endl;
```

```
// Rectangle
double length, width;
cout << "Enter the length of the rectangle: ";
cin >> length;
cout << "Enter the width of the rectangle: ";
cin >> width;

double rectangle_area = length * width;
double rectangle_perimeter = 2 * (length + width);

cout << "Area of the rectangle: " <<
rectangle_area << endl;
cout << "Perimeter of the rectangle: " <<
rectangle_perimeter << endl;

return 0;
}
```



Thanks

