Introduction

C++

From Wi the free encyclopedia

C++ (pronc plʌs plʌs/) is a general
programming g the facilities for lov

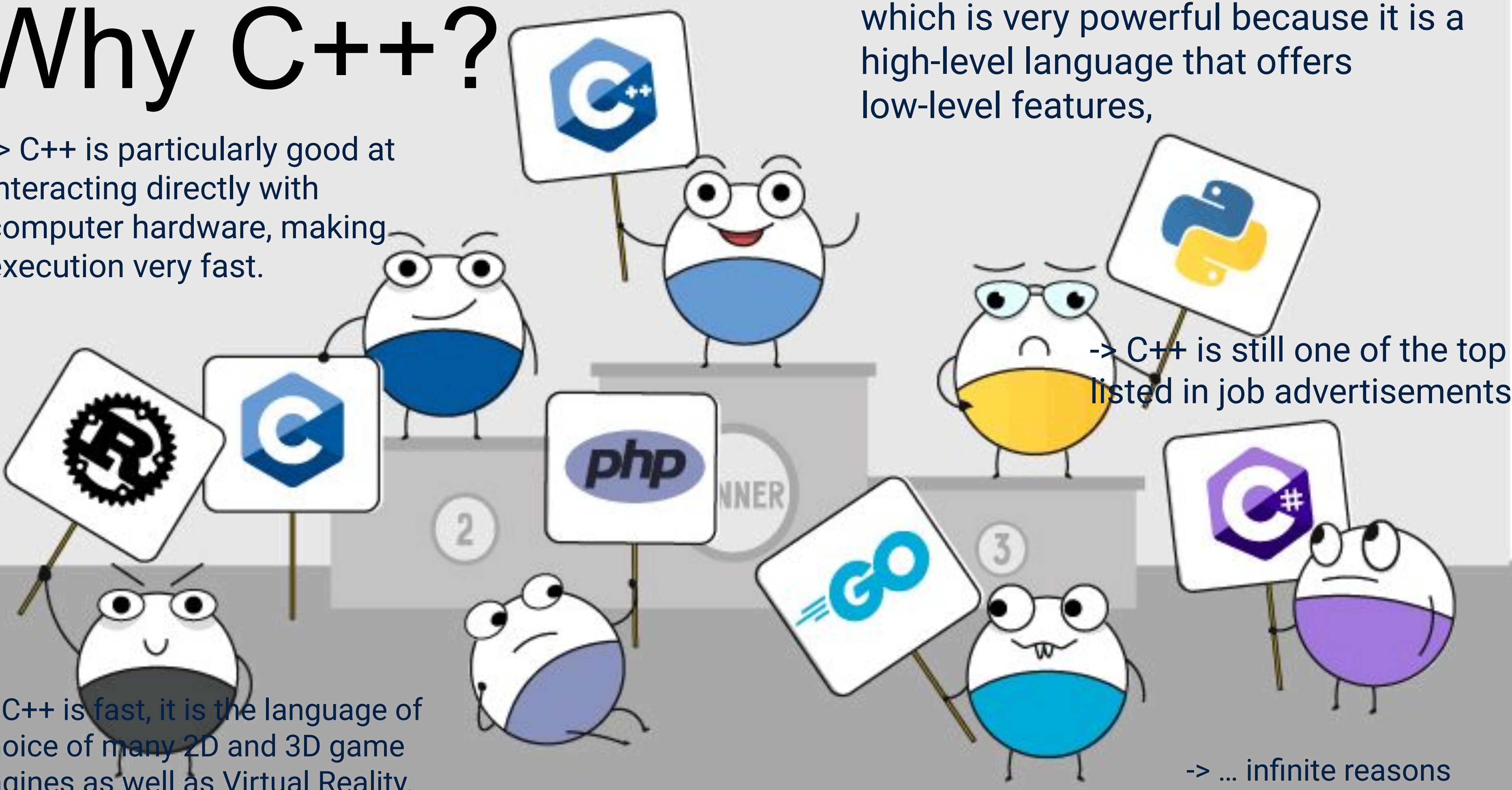
It is designed w ystem programming (e.g., fo
efficiency and fle esign requirements. C++
servers (e.g. e-cor we SQL servers), perfo
entertainment softwa C++ is a language, with im
organizations, includir he FSF, LLVI icrosoft and Intel.

C++ is standardized by the International Organization for Stan
published by ISO in December 2014 as *ISO/IEC 14882:2014* (
standardized in 1998 as *ISO/IEC 14882:1998*, which was then
standard supersedes these and C++11, with new

Why C++?

-> C++ is particularly good at interacting directly with computer hardware, making execution very fast.

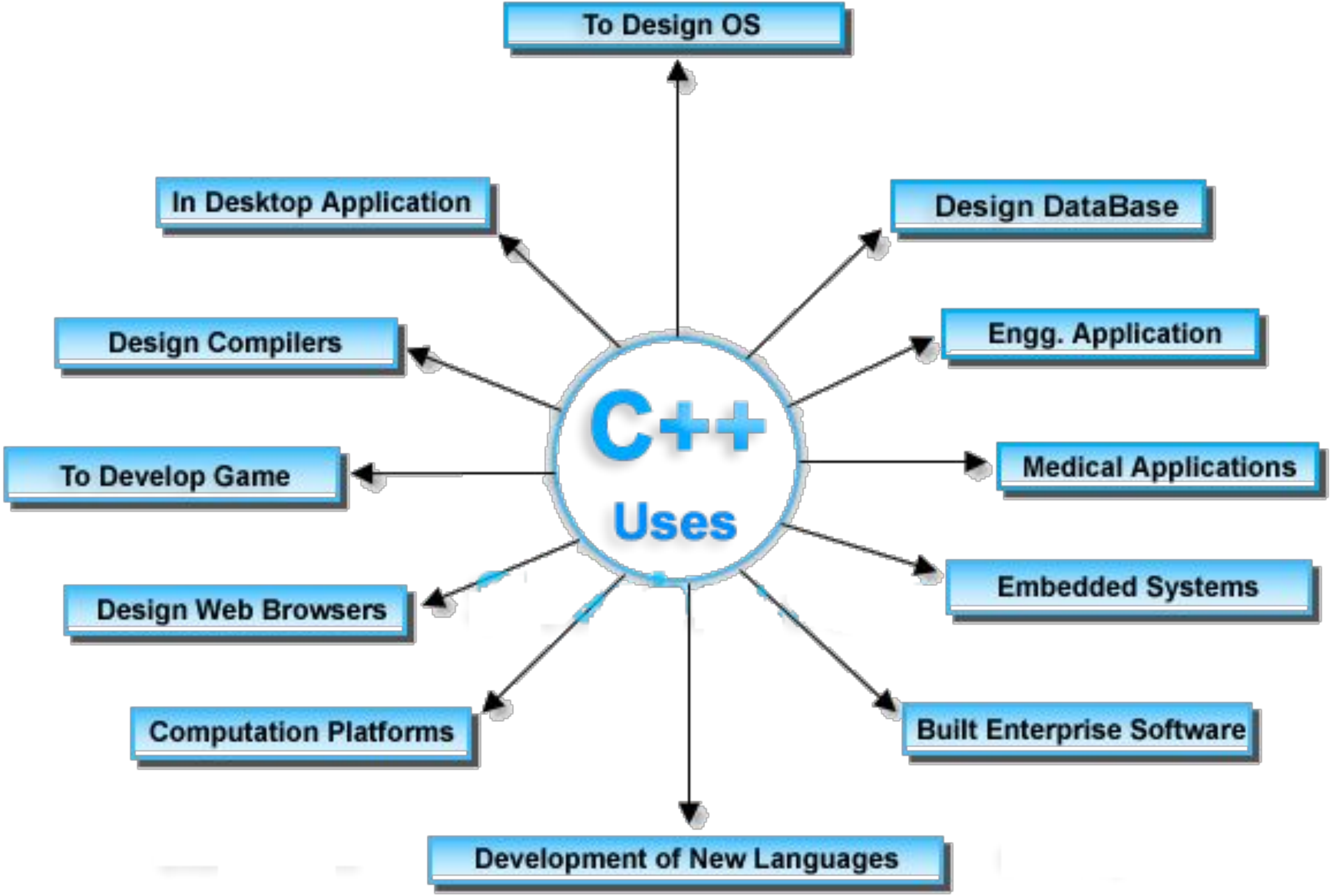
-> C++ is an enormous language which is very powerful because it is a high-level language that offers low-level features,

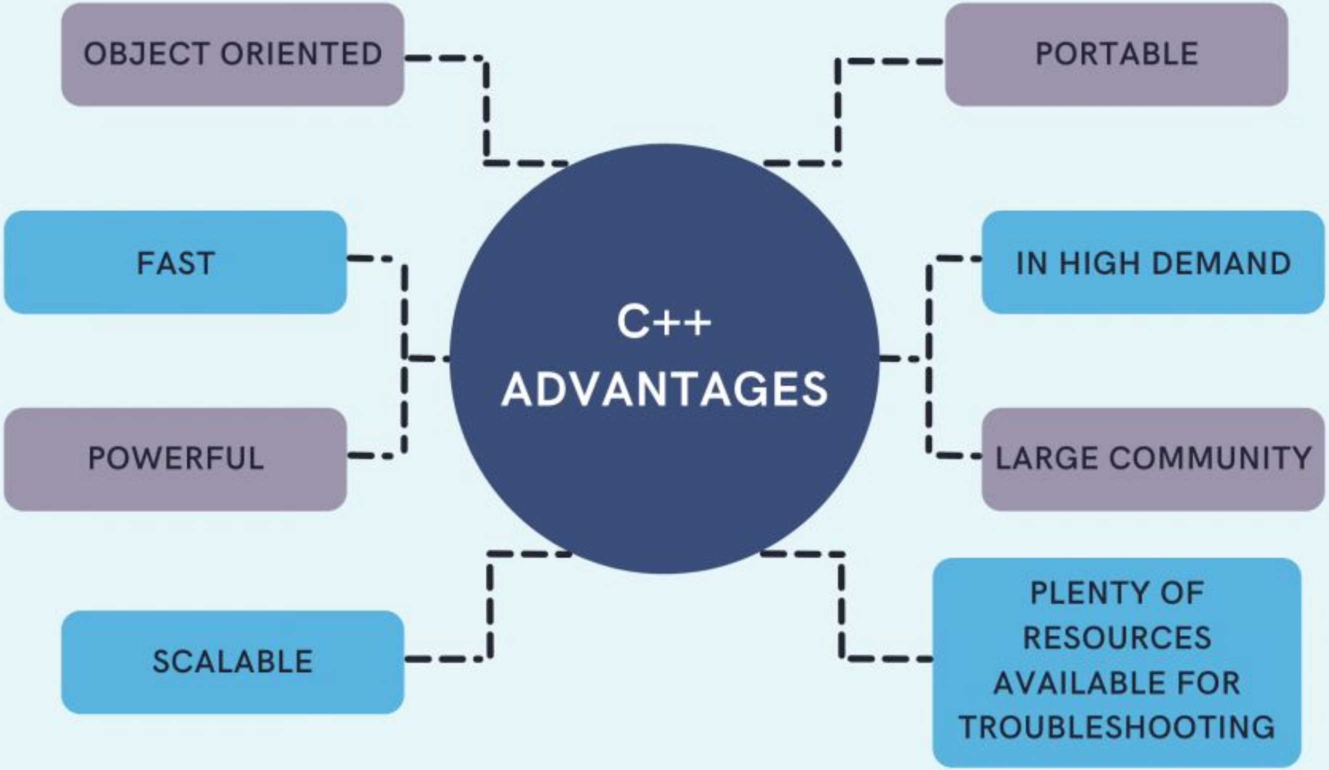


-> C++ is still one of the top listed in job advertisements

-> C++ is fast, it is the language of choice of many 2D and 3D game engines as well as Virtual Reality.

-> … infinite reasons


To Design OS
In Desktop Application
Design DataBase
Design Compilers
Engg. Application
C++
Uses
To Develop Game
Medical Applications
Design Web Browsers
Embedded Systems
Computation Platforms
Built Enterprise Software
Development of New Languages



C++ ADVANTAGES

OBJECT ORIENTED

FAST

POWERFUL

SCALABLE

PORTABLE

IN HIGH DEMAND

LARGE COMMUNITY

PLENTY OF RESOURCES AVAILABLE FOR TROUBLESHOOTING

Disadvantages:

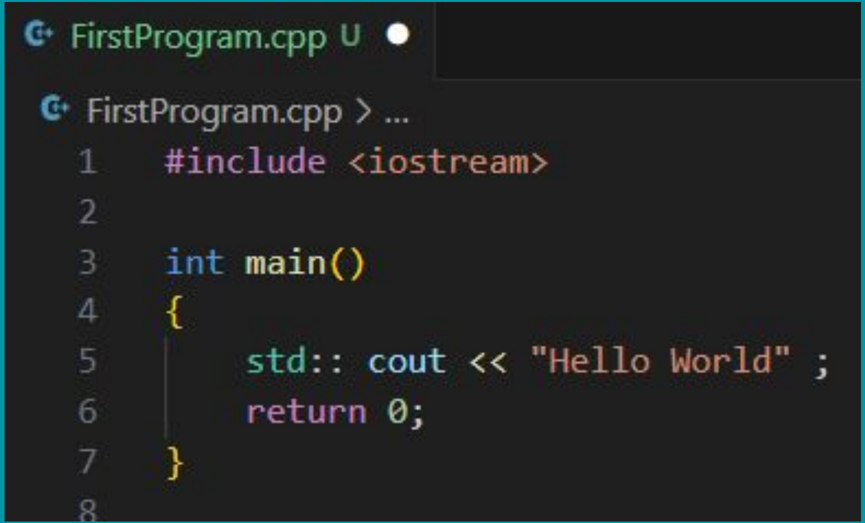
- No Disadvantages, it is the best language.

Disadvantages

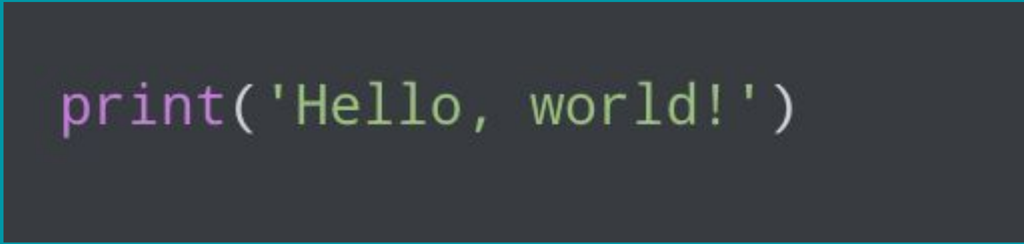
- Complexity: Extensive feature set leading to a steep learning curve.
- Manual Memory Management: Potential memory leaks and issues without proper handling.
- Lack of Garbage Collection: Manual deallocation of memory required.
- Complex Syntax: Time-consuming code reading and writing compared to other languages.
- others…

Installation with dev c++

Download only for windows users



```
1   #include <iostream>
2
3   int main()
4   {
5       std:: cout << "Hello World" ;
6       return 0;
7   }
8
```



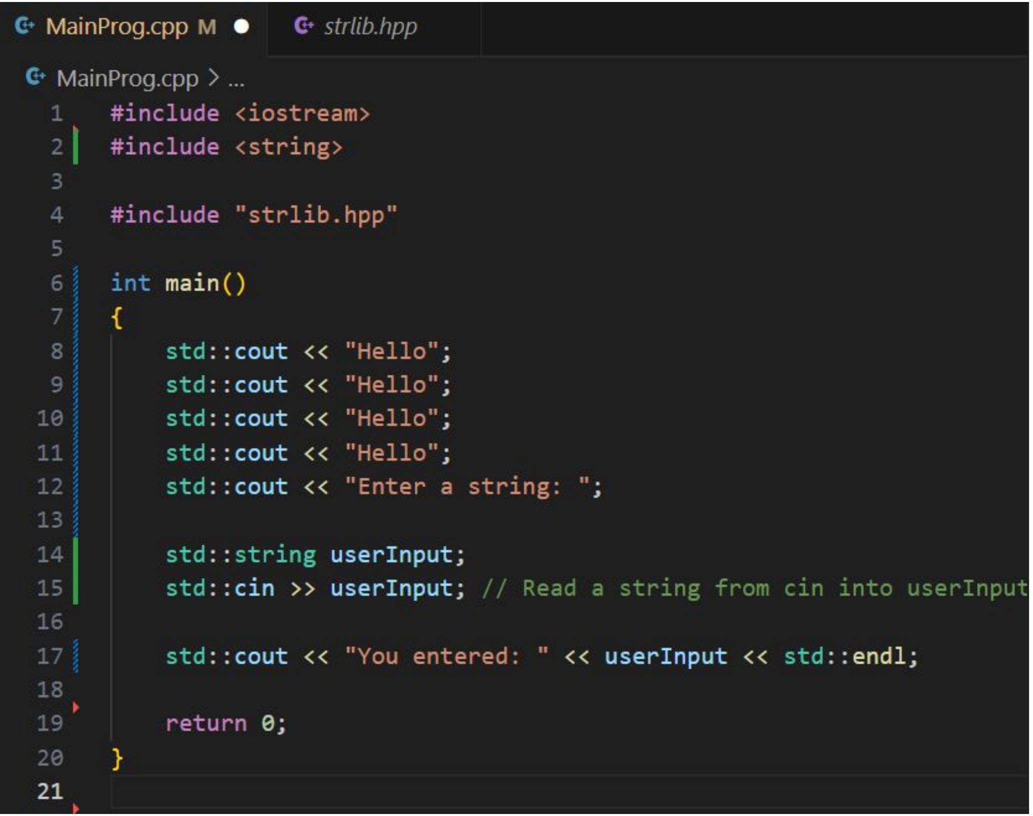
```
print('Hello, world!')
```

Line 1: `#include <iostream>` is a header file library that lets us work with input and output objects, such as `cout`

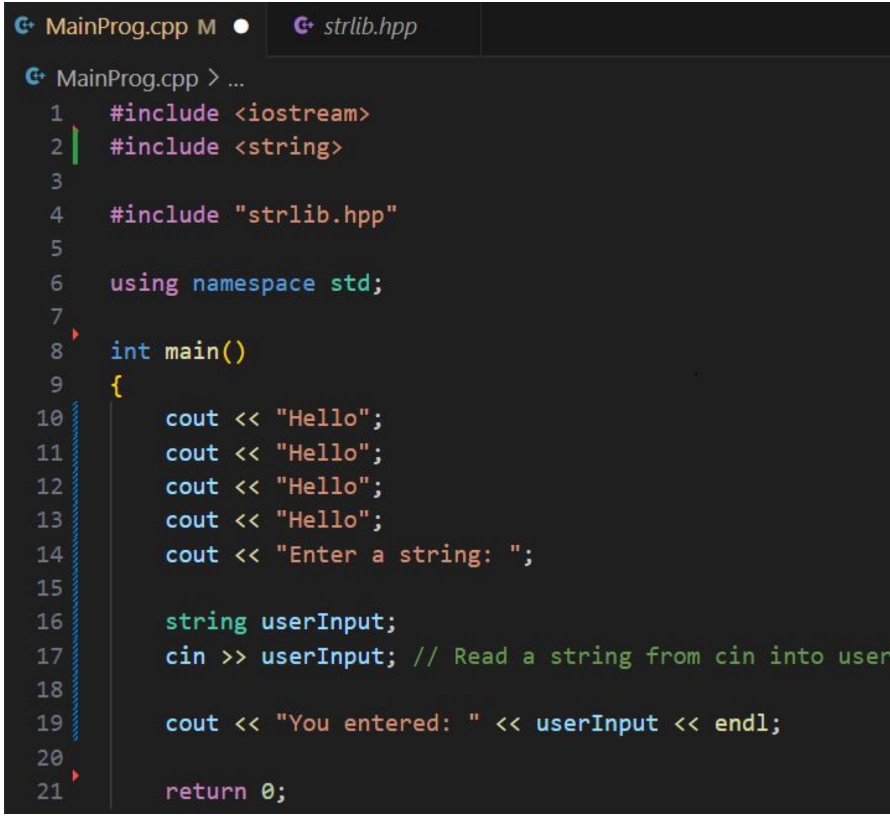
Line 3: Another thing that always appear in a C++ program is `int main()`. This is called a function. Any code inside its curly brackets `{}` will be executed.

Line 5: cout is an object in C++ that represents the standard output stream, usually the console. It is part of the std namespace

`cout` (pronounced "see-out") is used together with the *insertion operator* (`<<`) to output/print text.

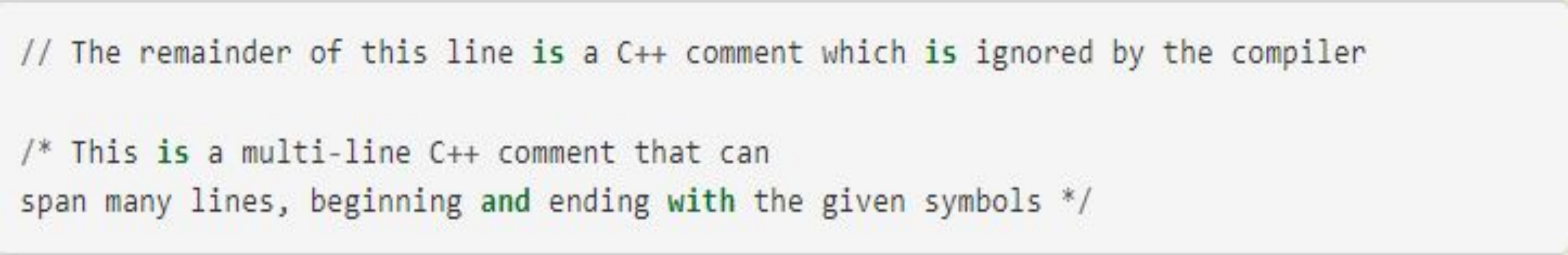


```cpp
#include <iostream>
#include <string>

#include "strlib.hpp"

int main()
{
    std::cout << "Hello";
    std::cout << "Hello";
    std::cout << "Hello";
    std::cout << "Hello";
    std::cout << "Enter a string: ";

    std::string userInput;
    std::cin >> userInput; // Read a string from cin into userInput

    std::cout << "You entered: " << userInput << std::endl;

    return 0;
}
```

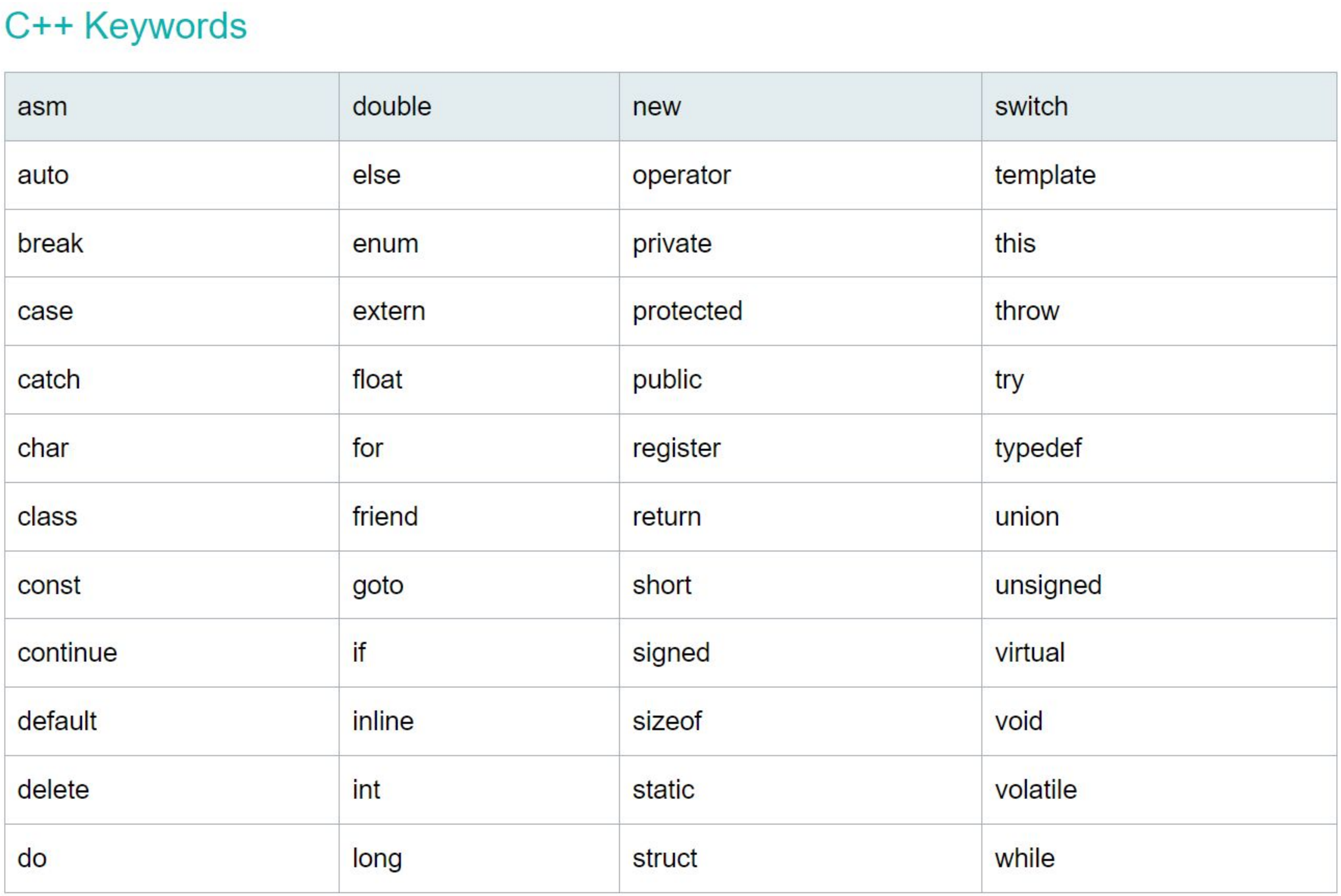


```cpp
#include <iostream>
#include <string>

#include "strlib.hpp"

using namespace std;

int main()
{
    cout << "Hello";
    cout << "Hello";
    cout << "Hello";
    cout << "Hello";
    cout << "Enter a string: ";

    string userInput;
    cin >> userInput; // Read a string from cin into user

    cout << "You entered: " << userInput << endl;

    return 0;
```

Comments in C++



```
// The remainder of this line is a C++ comment which is ignored by the compiler

/* This is a multi-line C++ comment that can
span many lines, beginning and ending with the given symbols */
```



C++ Keywords

asm	double	new	switch
auto	else	operator	template
break	enum	private	this
case	extern	protected	throw
catch	float	public	try
char	for	register	typedef
class	friend	return	union
const	goto	short	unsigned
continue	if	signed	virtual
default	inline	sizeof	void
delete	int	static	volatile
do	long	struct	while

Data type in C++

Primitive

-> int

-> float

-> double

-> bool

-> **char**

Derived

-> Function

-> Array

-> Vector

-> **Pointer**

-> **Reference**

-> String

User-Defined

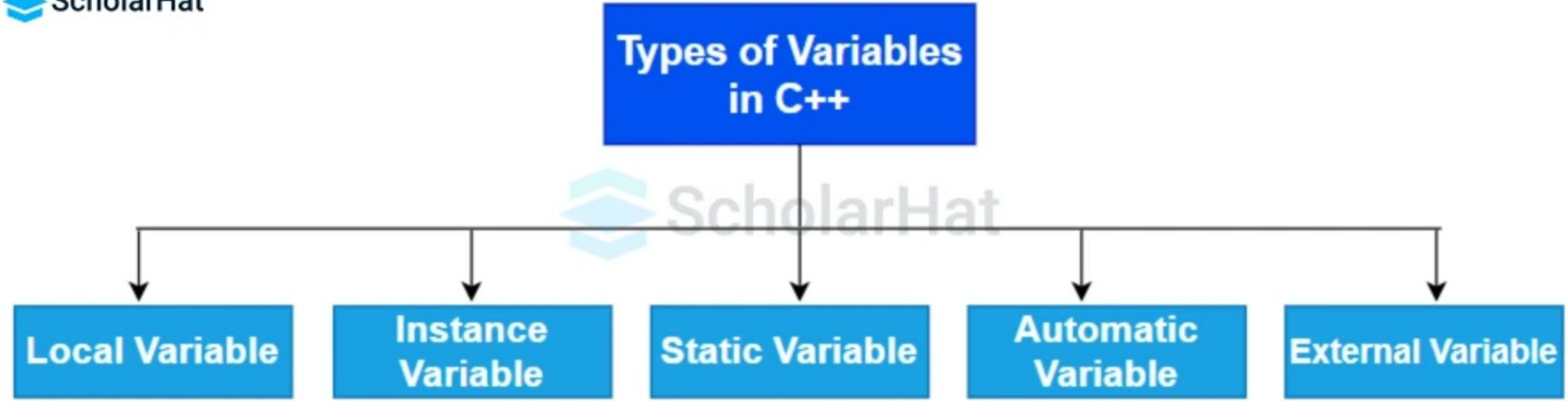
-> **Class**

-> **Structure**

-> Union

-> **Enumeration**

-> TypeDef



Types of Variables in C++

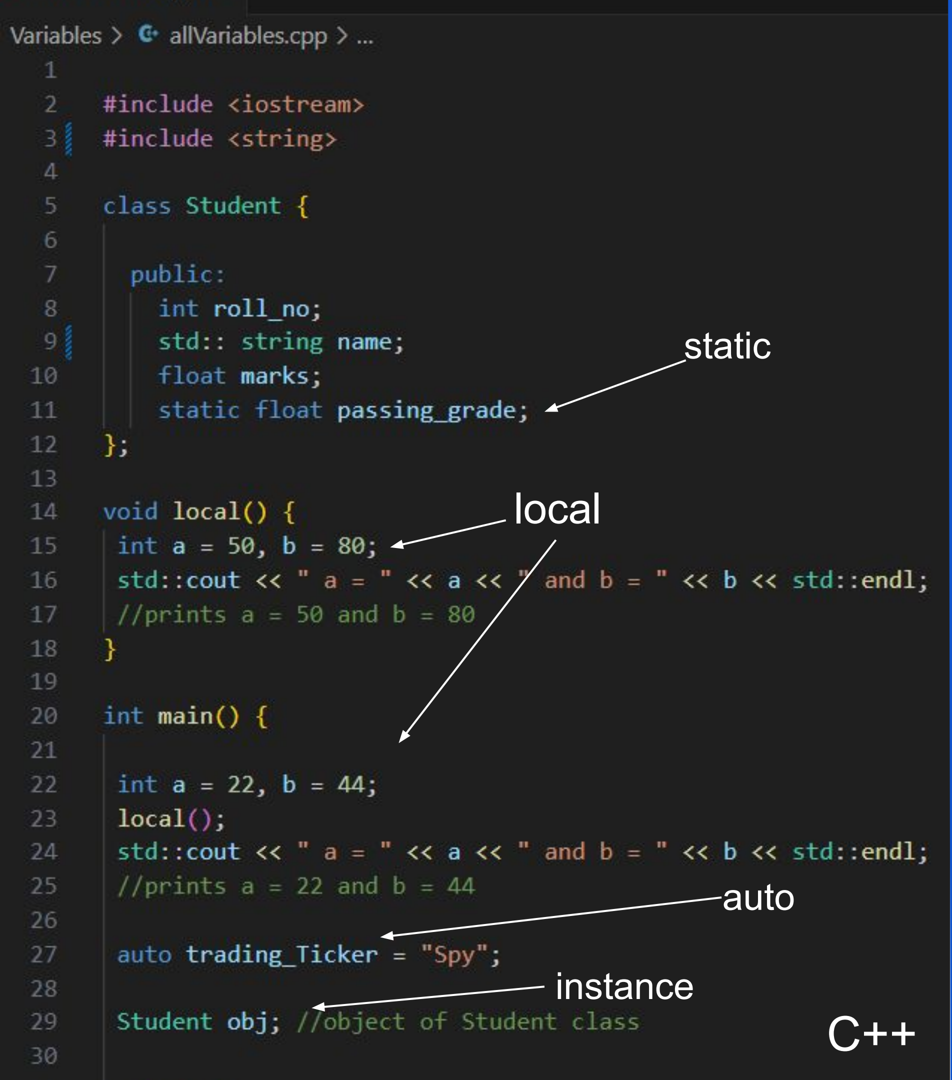
Local Variable

Instance Variable

Static Variable

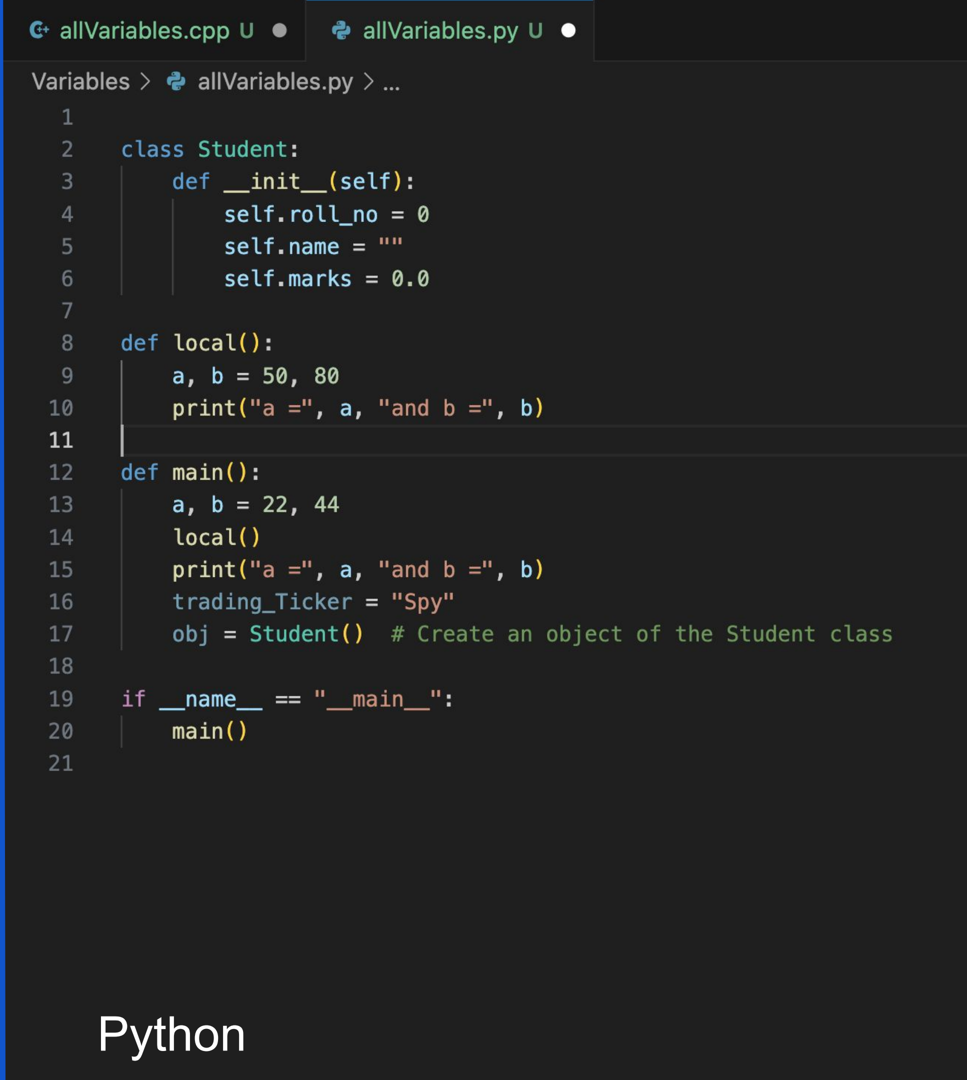
Automatic Variable

External Variable



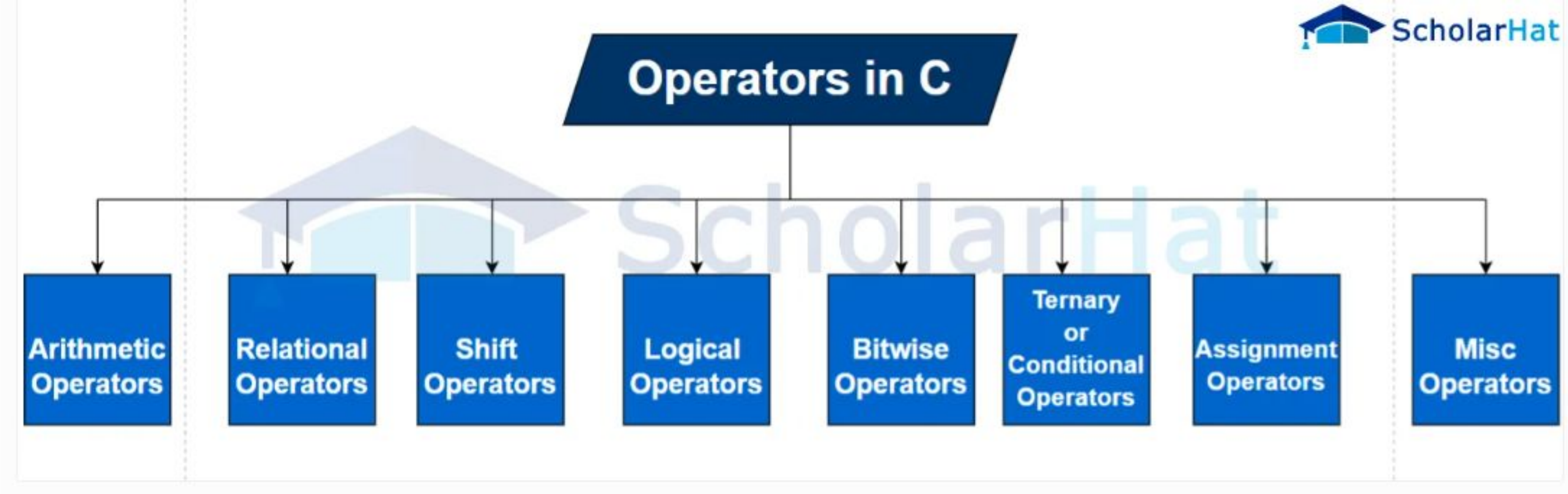
```cpp

#include <iostream>
#include <string>

class Student {

  public:
    int roll_no;
    std:: string name;
    float marks;
    static float passing_grade;
};

void local() {
 int a = 50, b = 80;
 std::cout << " a = " << a << " and b = " << b << std::endl;
 //prints a = 50 and b = 80
}

int main() {

 int a = 22, b = 44;
 local();
 std::cout << " a = " << a << " and b = " << b << std::endl;
 //prints a = 22 and b = 44

 auto trading_Ticker = "Spy";

 Student obj; //object of Student class

```

C++

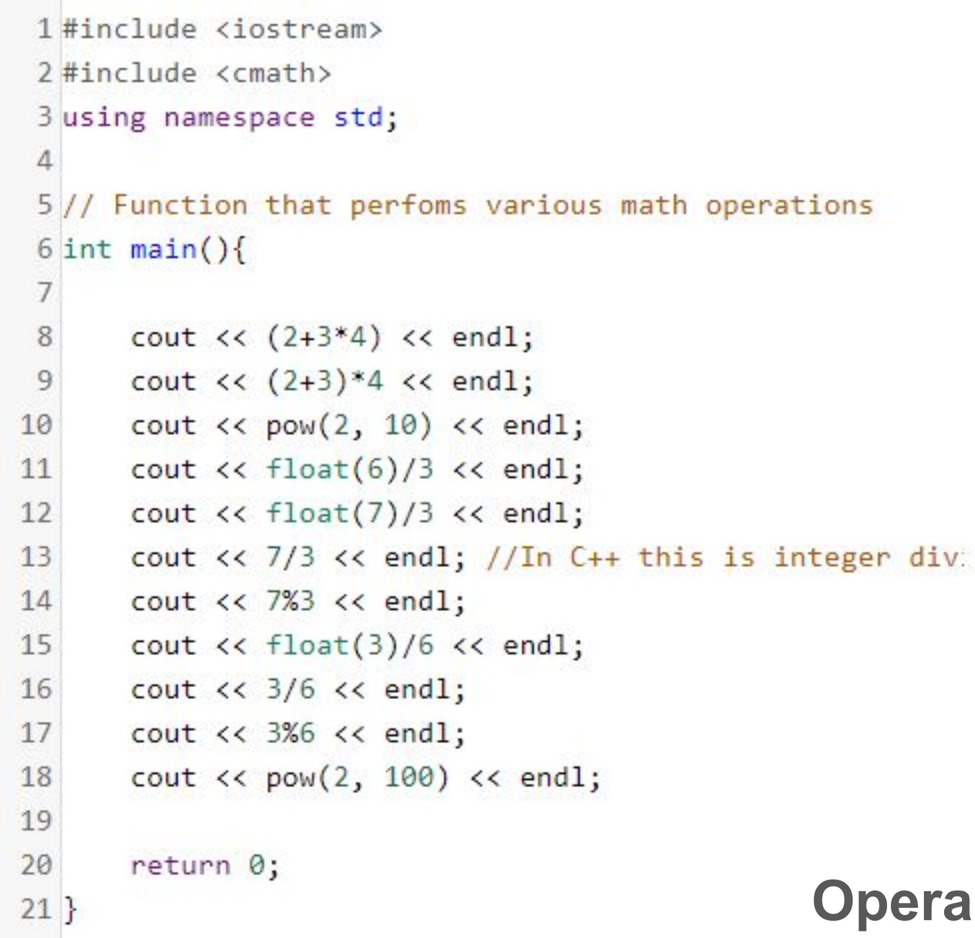


```python

class Student:
    def __init__(self):
        self.roll_no = 0
        self.name = ""
        self.marks = 0.0

def local():
    a, b = 50, 80
    print("a =", a, "and b =", b)

def main():
    a, b = 22, 44
    local()
    print("a =", a, "and b =", b)
    trading_Ticker = "Spy"
    obj = Student()  # Create an object of the Student class

if __name__ == "__main__":
    main()

```

Python

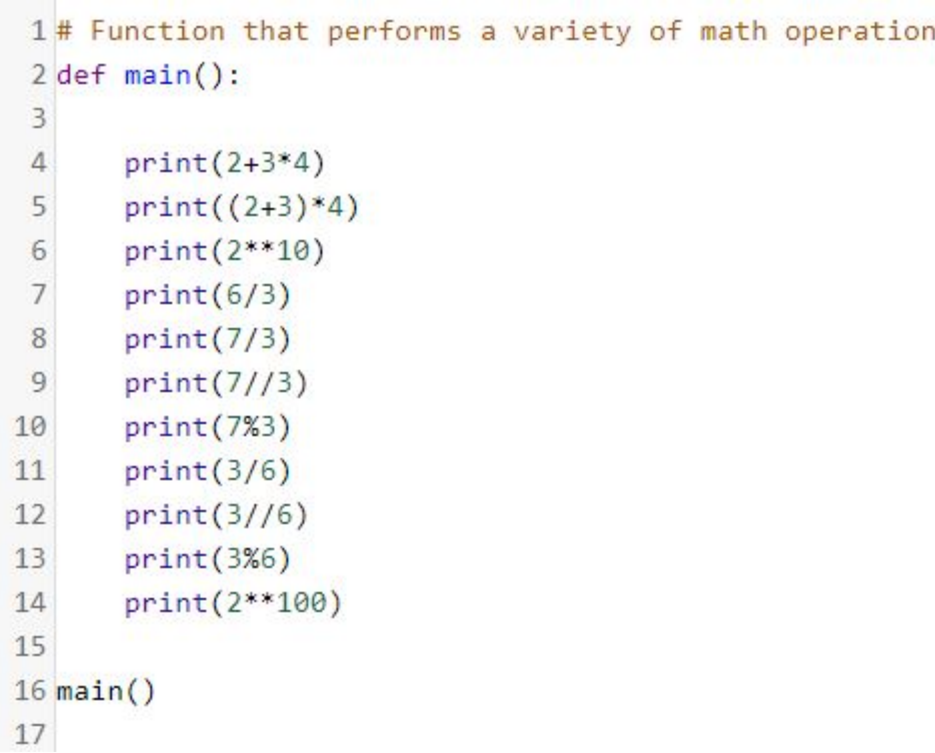

Operators in C
Arithmetic Operators
Relational Operators
Shift Operators
Logical Operators
Bitwise Operators
Ternary or Conditional Operators
Assignment Operators
Misc Operators
ScholarHat

Operators



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

// Function that perfoms various math operations
int main(){

    cout << (2+3*4) << endl;
    cout << (2+3)*4 << endl;
    cout << pow(2, 10) << endl;
    cout << float(6)/3 << endl;
    cout << float(7)/3 << endl;
    cout << 7/3 << endl; //In C++ this is integer div
    cout << 7%3 << endl;
    cout << float(3)/6 << endl;
    cout << 3/6 << endl;
    cout << 3%6 << endl;
    cout << pow(2, 100) << endl;

    return 0;
}
```

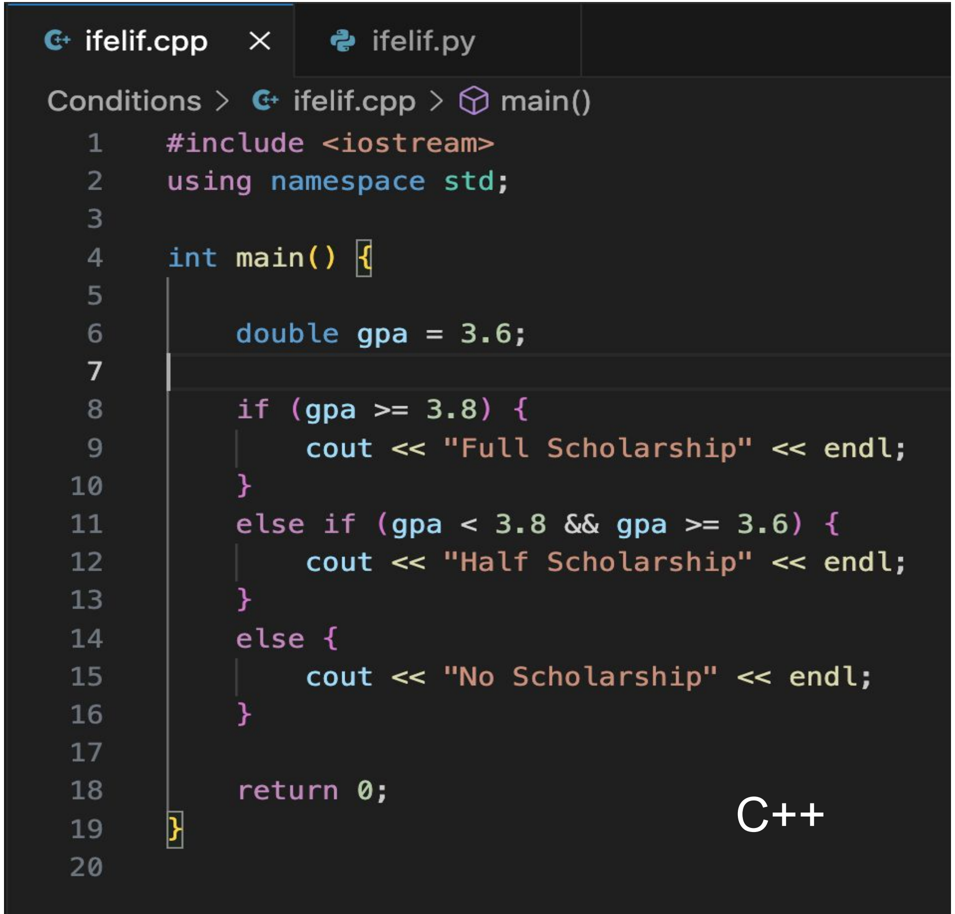
C++



```python
# Function that performs a variety of math operation
def main():

    print(2+3*4)
    print((2+3)*4)
    print(2**10)
    print(6/3)
    print(7/3)
    print(7//3)
    print(7%3)
    print(3/6)
    print(3//6)
    print(3%6)
    print(2**100)

main()
```

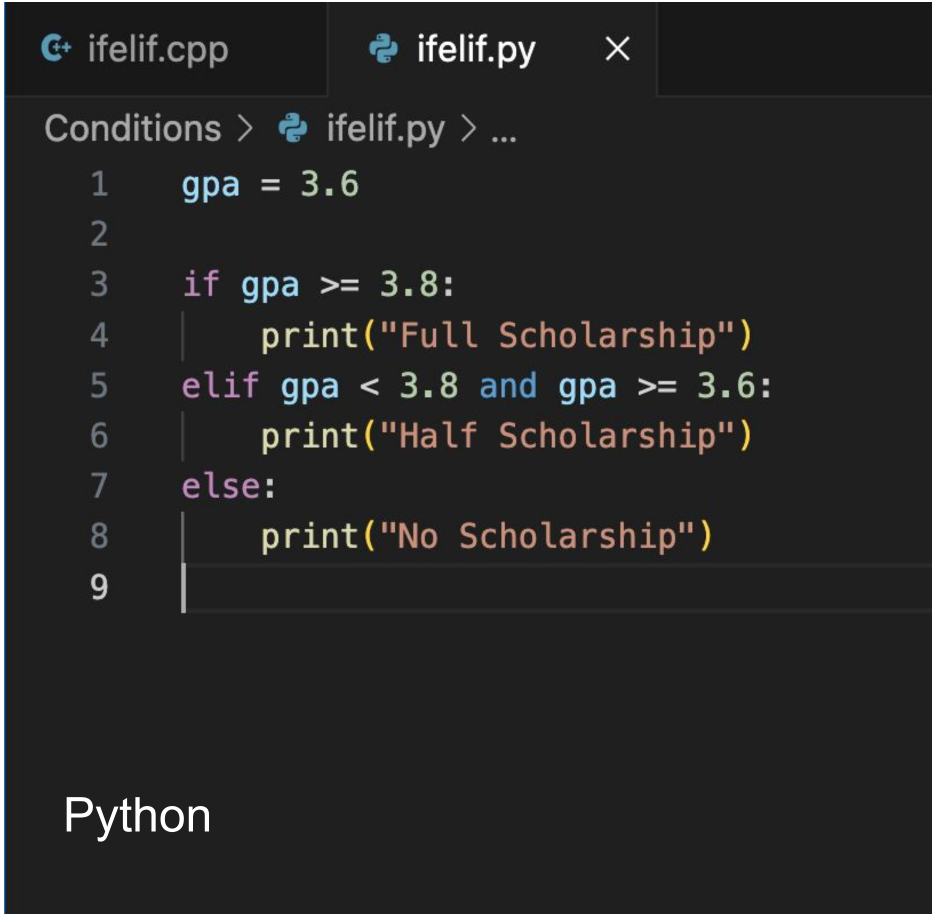
Python

Conditional Statement, if/else



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

int main() {

    double gpa = 3.6;

    if (gpa >= 3.8) {
        cout << "Full Scholarship" << endl;
    }
    else if (gpa < 3.8 && gpa >= 3.6) {
        cout << "Half Scholarship" << endl;
    }
    else {
        cout << "No Scholarship" << endl;
    }

    return 0;
}
```

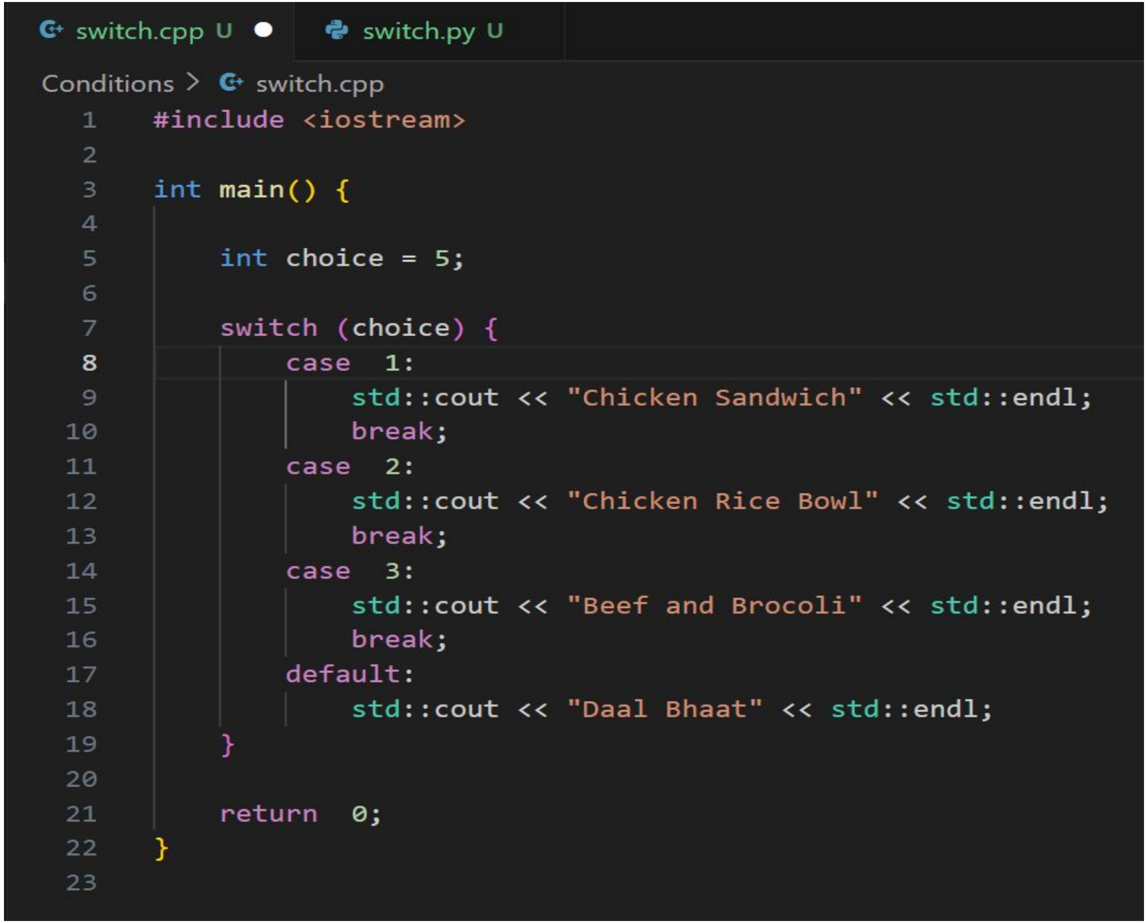
C++



```python
gpa = 3.6

if gpa >= 3.8:
    print("Full Scholarship")
elif gpa < 3.8 and gpa >= 3.6:
    print("Half Scholarship")
else:
    print("No Scholarship")
```

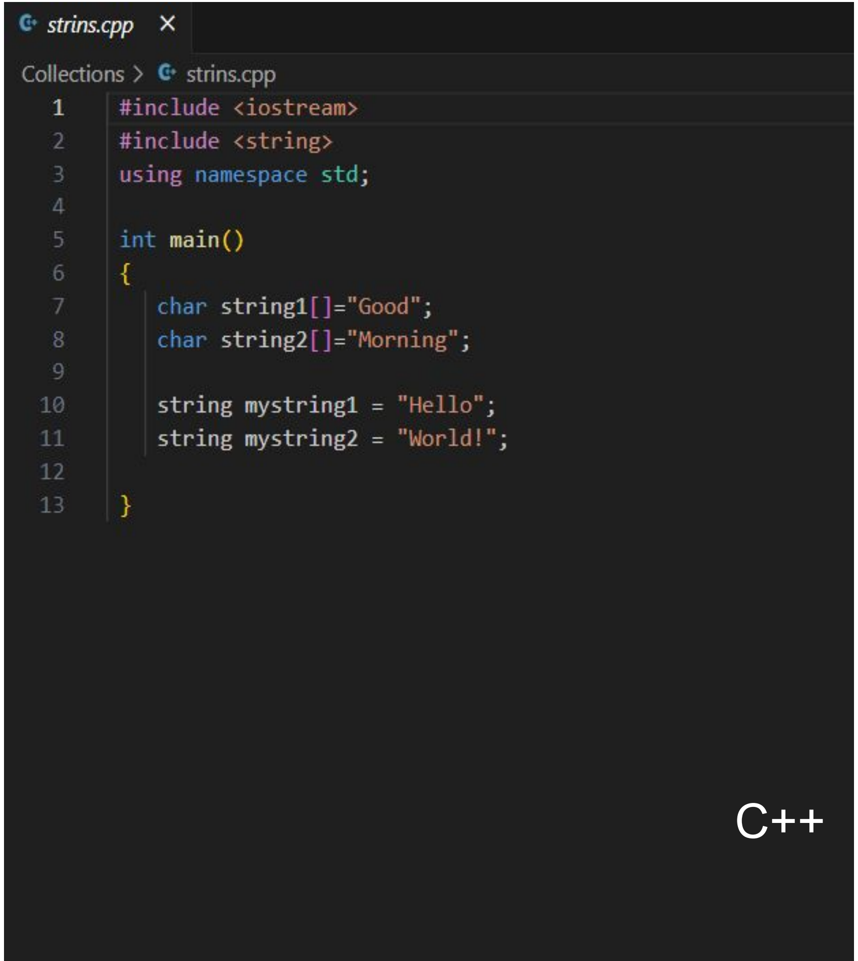
Python

Switch Statement



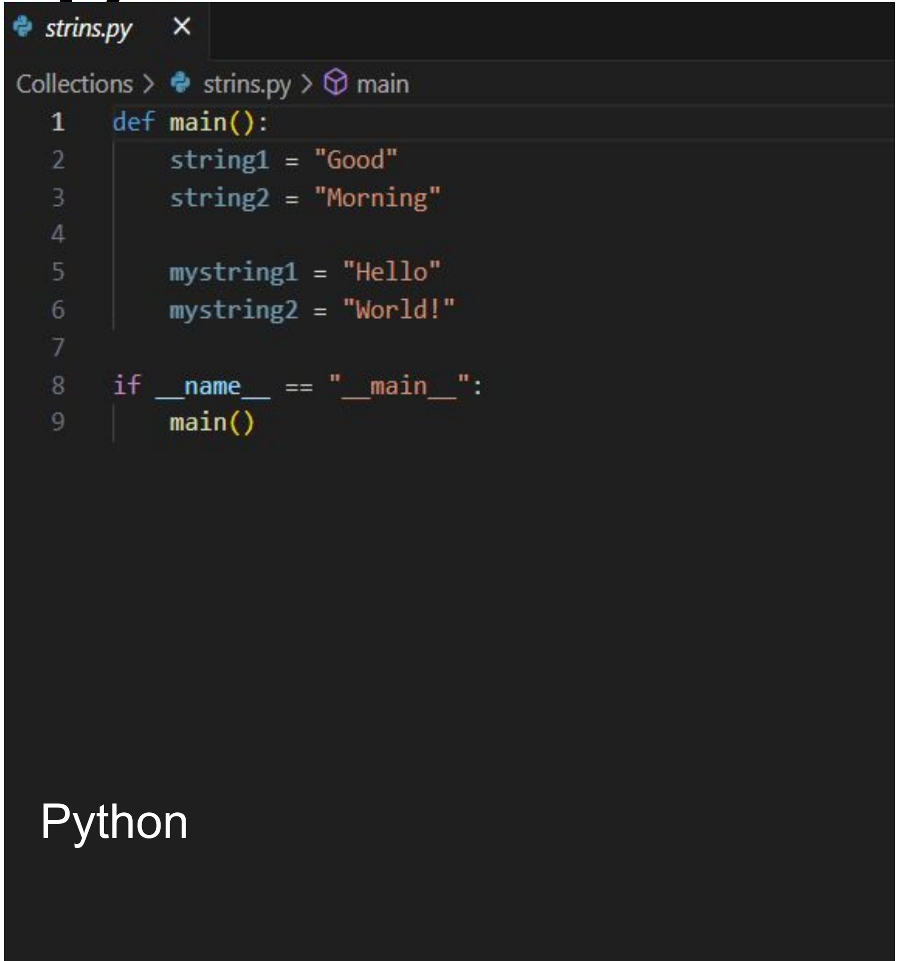
```cpp
#include <iostream>

int main() {

    int choice = 5;

    switch (choice) {
        case  1:
            std::cout << "Chicken Sandwich" << std::endl;
            break;
        case  2:
            std::cout << "Chicken Rice Bowl" << std::endl;
            break;
        case  3:
            std::cout << "Beef and Brocoli" << std::endl;
            break;
        default:
            std::cout << "Daal Bhaat" << std::endl;
    }

    return  0;
}

```

Strings



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

int main()
{
    char string1[]="Good";
    char string2[]="Morning";

    string mystring1 = "Hello";
    string mystring2 = "World!";

}
```

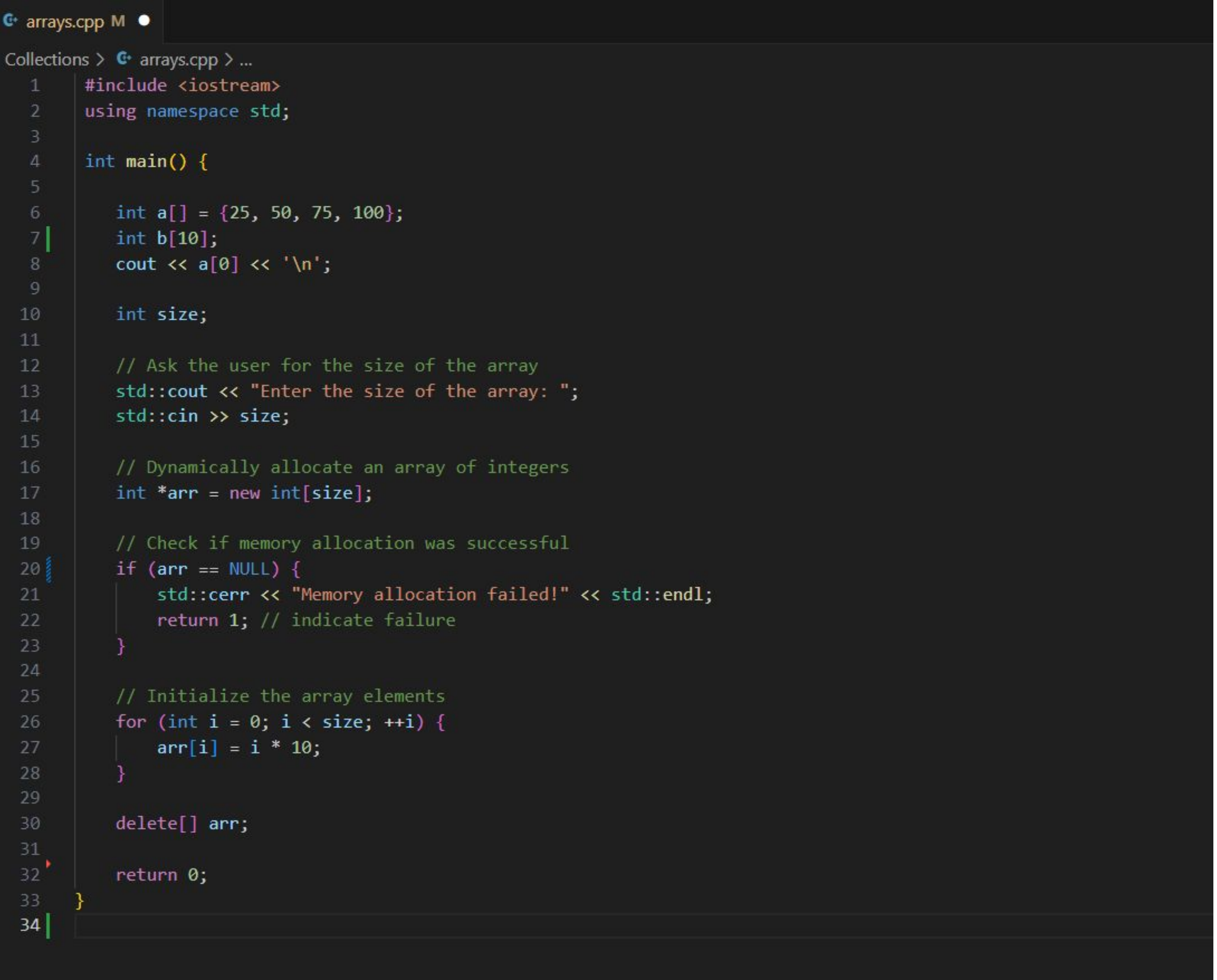
C++



```python
def main():
    string1 = "Good"
    string2 = "Morning"

    mystring1 = "Hello"
    mystring2 = "World!"

if __name__ == "__main__":
    main()
```

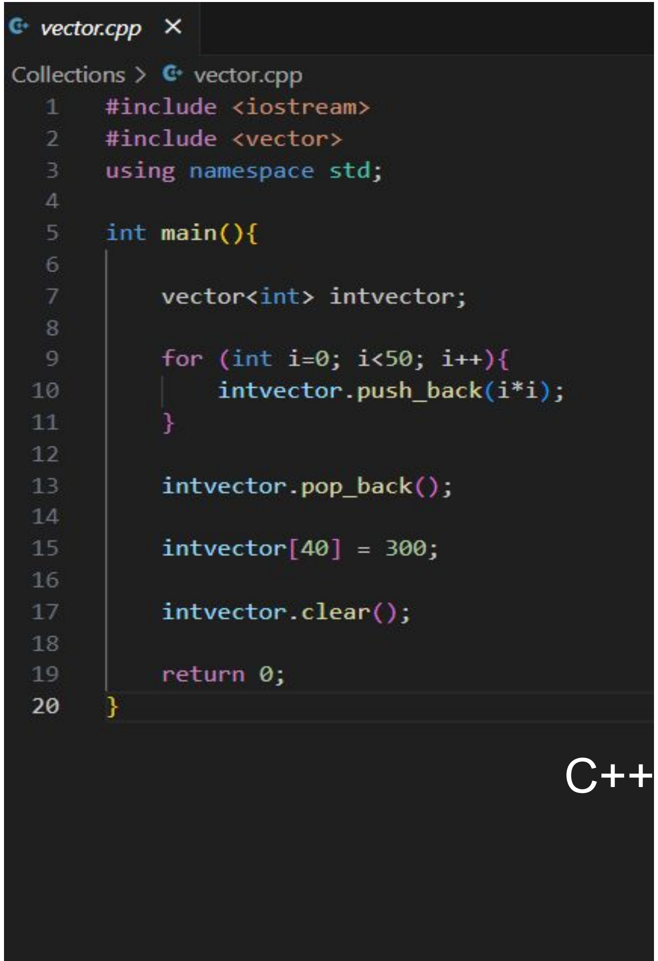
Python

Arrays



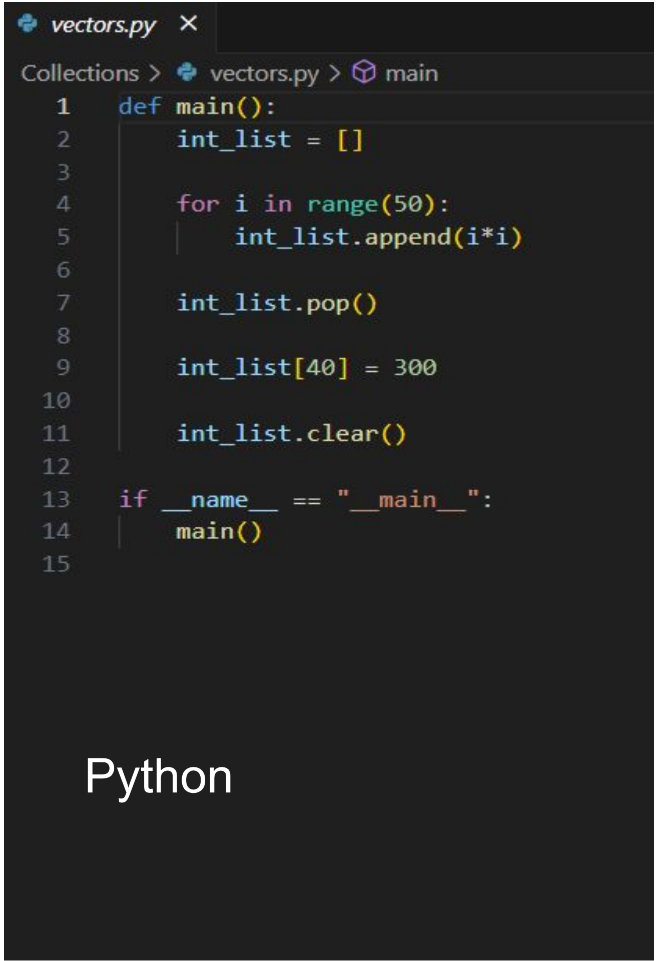
```cpp
#include <iostream>
using namespace std;

int main() {

    int a[] = {25, 50, 75, 100};
    int b[10];
    cout << a[0] << '\n';

    int size;

    // Ask the user for the size of the array
    std::cout << "Enter the size of the array: ";
    std::cin >> size;

    // Dynamically allocate an array of integers
    int *arr = new int[size];

    // Check if memory allocation was successful
    if (arr == NULL) {
        std::cerr << "Memory allocation failed!" << std::endl;
        return 1; // indicate failure
    }

    // Initialize the array elements
    for (int i = 0; i < size; ++i) {
        arr[i] = i * 10;
    }

    delete[] arr;

    return 0;
}

```

Vectors



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

int main(){

    vector<int> intvector;

    for (int i=0; i<50; i++){
        intvector.push_back(i*i);
    }

    intvector.pop_back();

    intvector[40] = 300;

    intvector.clear();

    return 0;
}
```

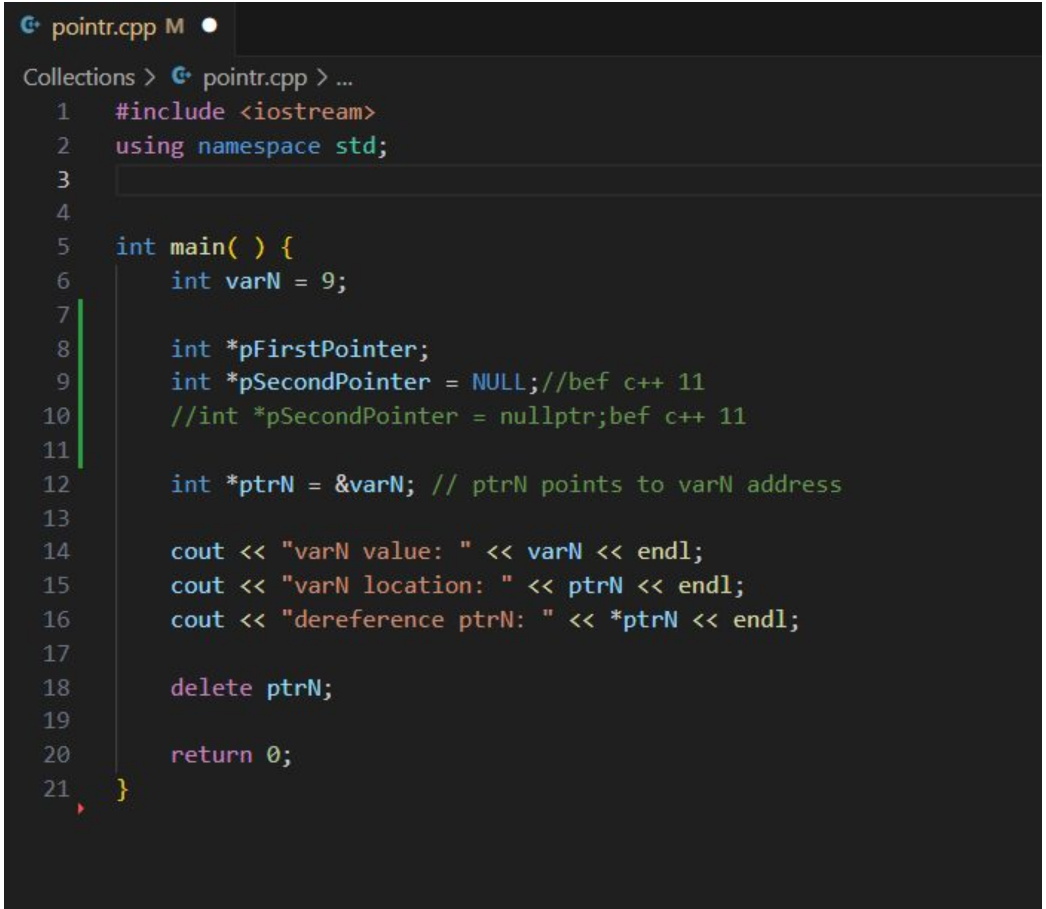
C++



```python
def main():
    int_list = []

    for i in range(50):
        int_list.append(i*i)

    int_list.pop()

    int_list[40] = 300

    int_list.clear()

if __name__ == "__main__":
    main()
```

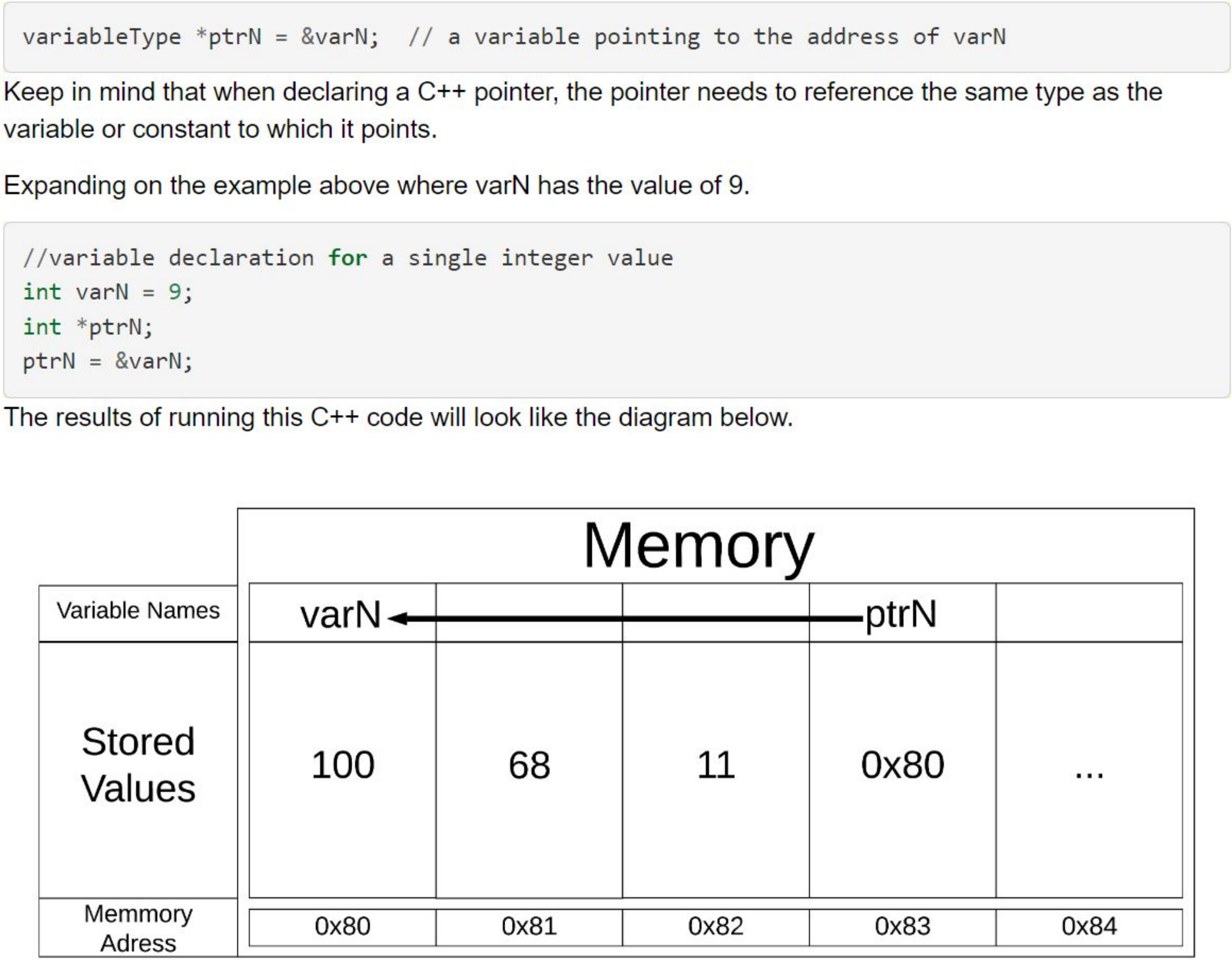
Python

Pointer



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


int main( ) {
    int varN = 9;

    int *pFirstPointer;
    int *pSecondPointer = NULL;//bef c++ 11
    //int *pSecondPointer = nullptr;bef c++ 11

    int *ptrN = &varN; // ptrN points to varN address

    cout << "varN value: " << varN << endl;
    cout << "varN location: " << ptrN << endl;
    cout << "dereference ptrN: " << *ptrN << endl;

    delete ptrN;

    return 0;
}
```

Pointer



```
variableType *ptrN = &varN;  // a variable pointing to the address of varN
```

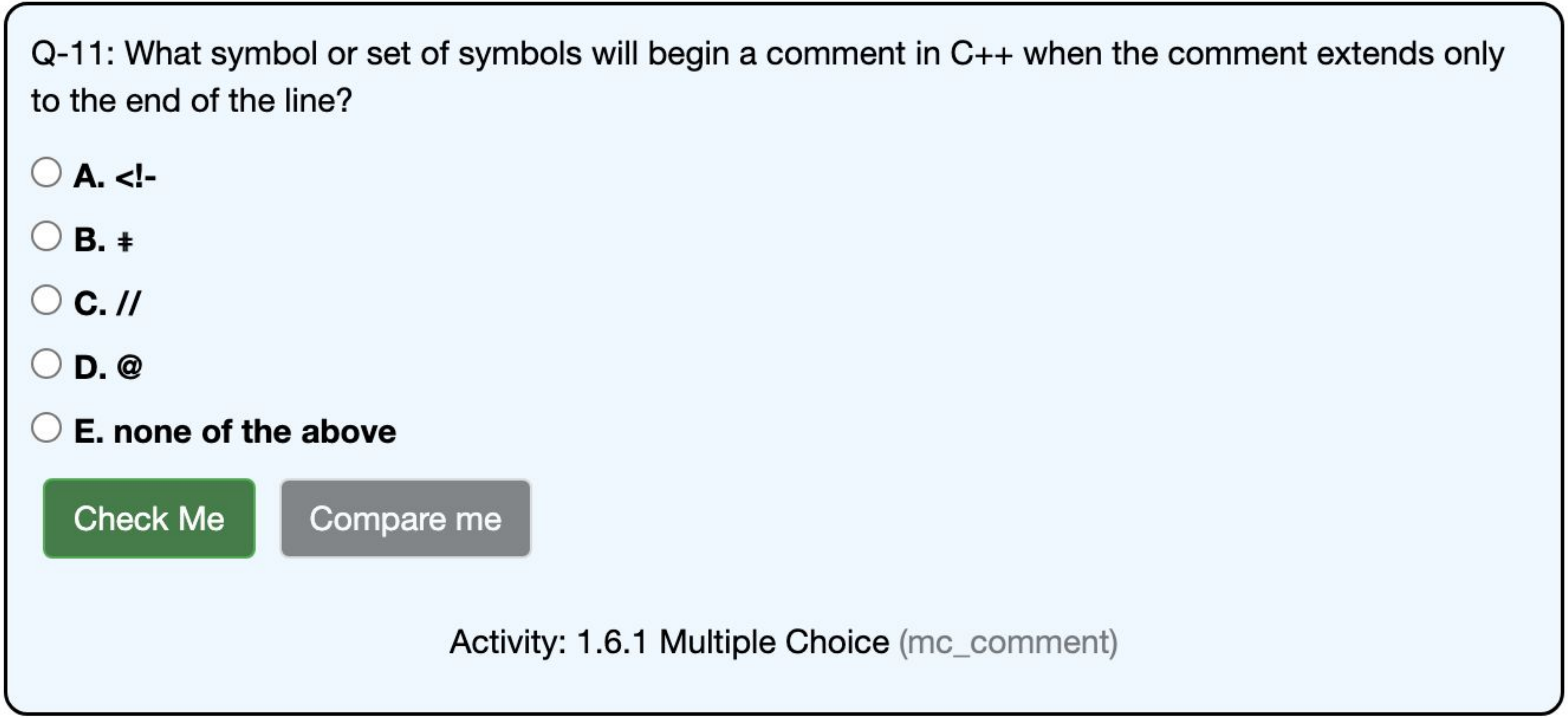
Keep in mind that when declaring a C++ pointer, the pointer needs to reference the same type as the variable or constant to which it points.

Expanding on the example above where varN has the value of 9.

```
//variable declaration for a single integer value
int varN = 9;
int *ptrN;
ptrN = &varN;
```

The results of running this C++ code will look like the diagram below.

	Memory				
Variable Names	varN ←			ptrN	
Stored Values	100	68	11	0x80	...
Memmory Adress	0x80	0x81	0x82	0x83	0x84



Q-11: What symbol or set of symbols will begin a comment in C++ when the comment extends only to the end of the line?

○ **A. <!-**

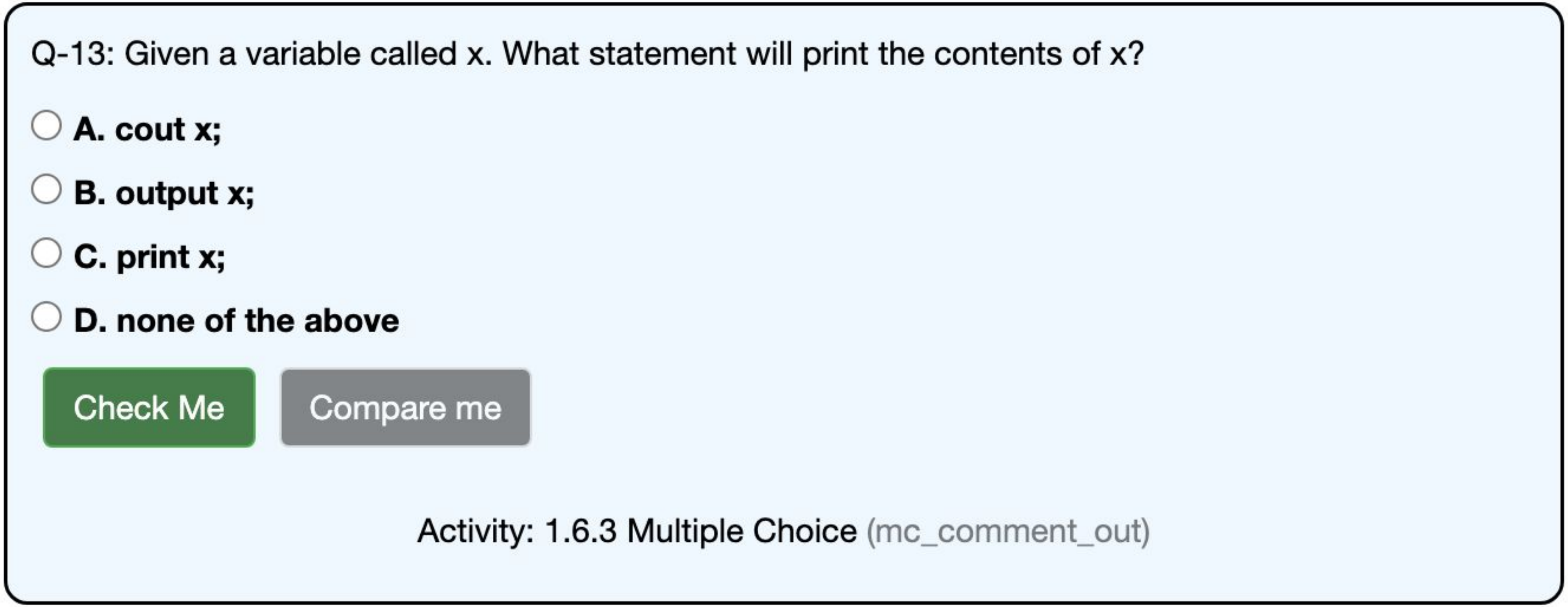
○ **B. #**

○ **C. //**

○ **D. @**

○ **E. none of the above**

Activity: 1.6.1 Multiple Choice (mc_comment)



Q-13: Given a variable called x. What statement will print the contents of x?

○ **A. cout x;**

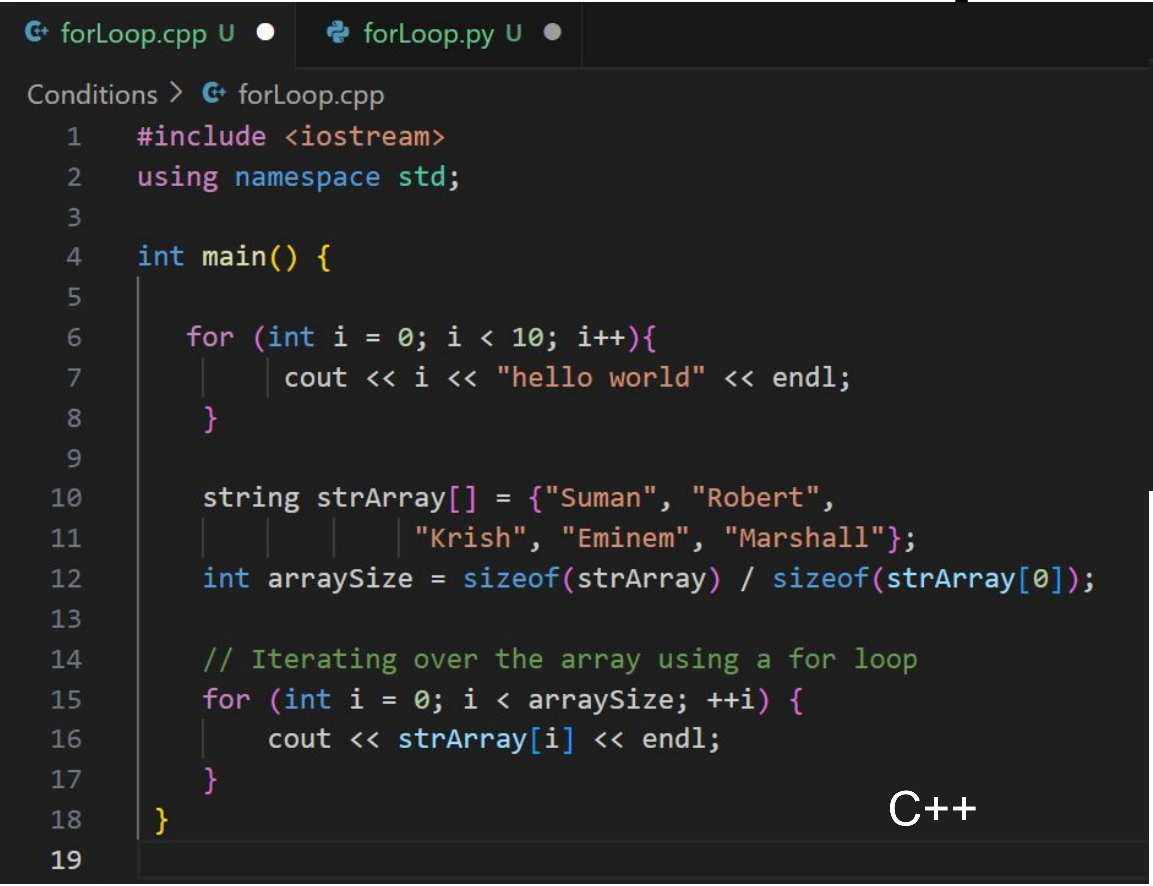
○ **B. output x;**

○ **C. print x;**

○ **D. none of the above**

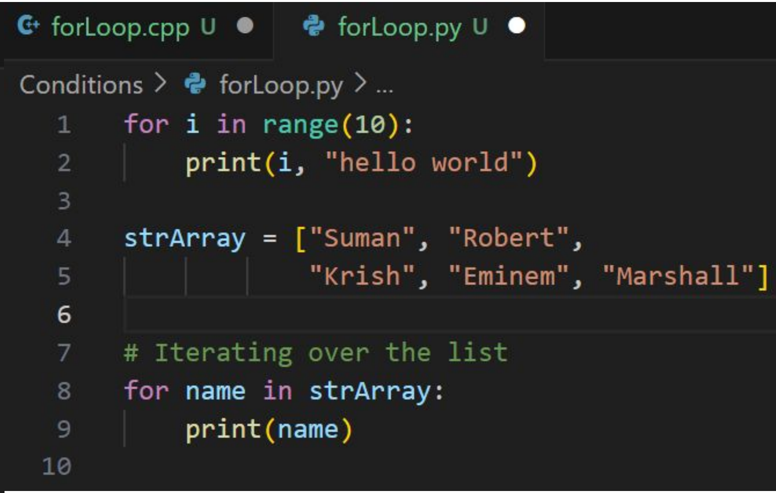
Activity: 1.6.3 Multiple Choice (mc_comment_out)

For Loop in C++



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

int main() {

   for (int i = 0; i < 10; i++){
         cout << i << "hello world" << endl;
   }

    string strArray[] = {"Suman", "Robert",
                         "Krish", "Eminem", "Marshall"};
    int arraySize = sizeof(strArray) / sizeof(strArray[0]);

    // Iterating over the array using a for loop
    for (int i = 0; i < arraySize; ++i) {
        cout << strArray[i] << endl;
    }
}
```

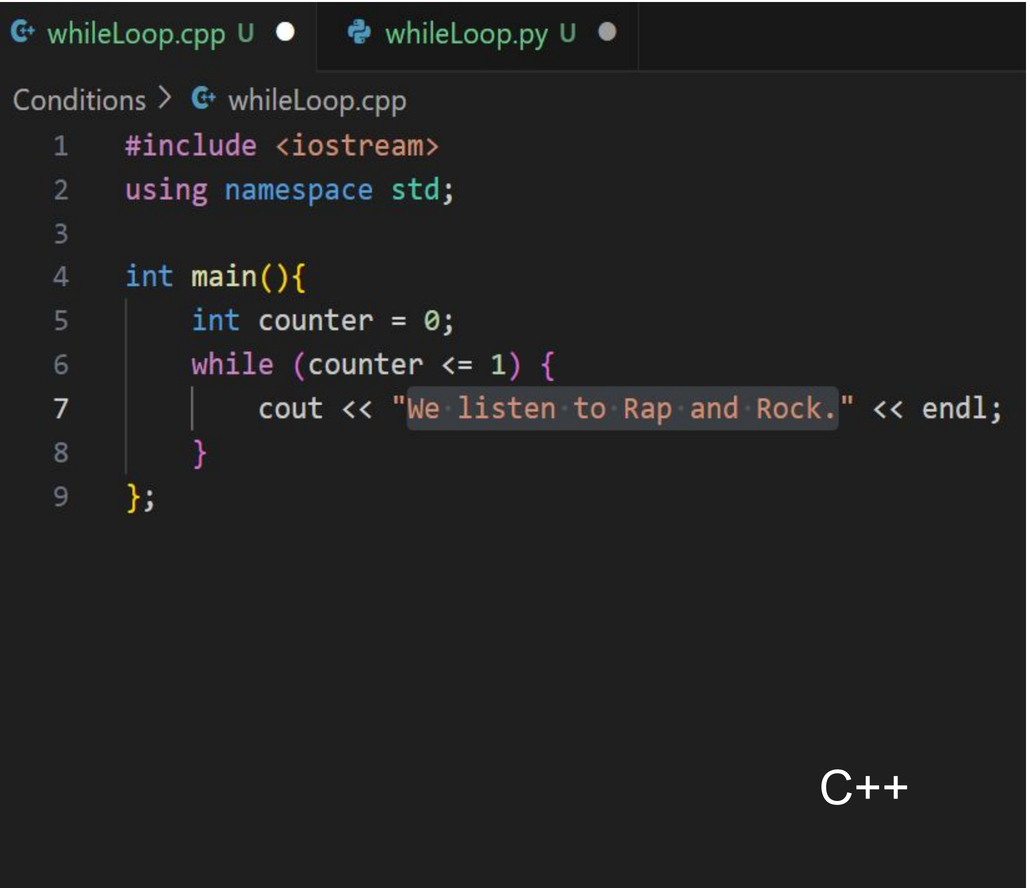
C++



```python
for i in range(10):
    print(i, "hello world")

strArray = ["Suman", "Robert",
            "Krish", "Eminem", "Marshall"]

# Iterating over the list
for name in strArray:
    print(name)

```

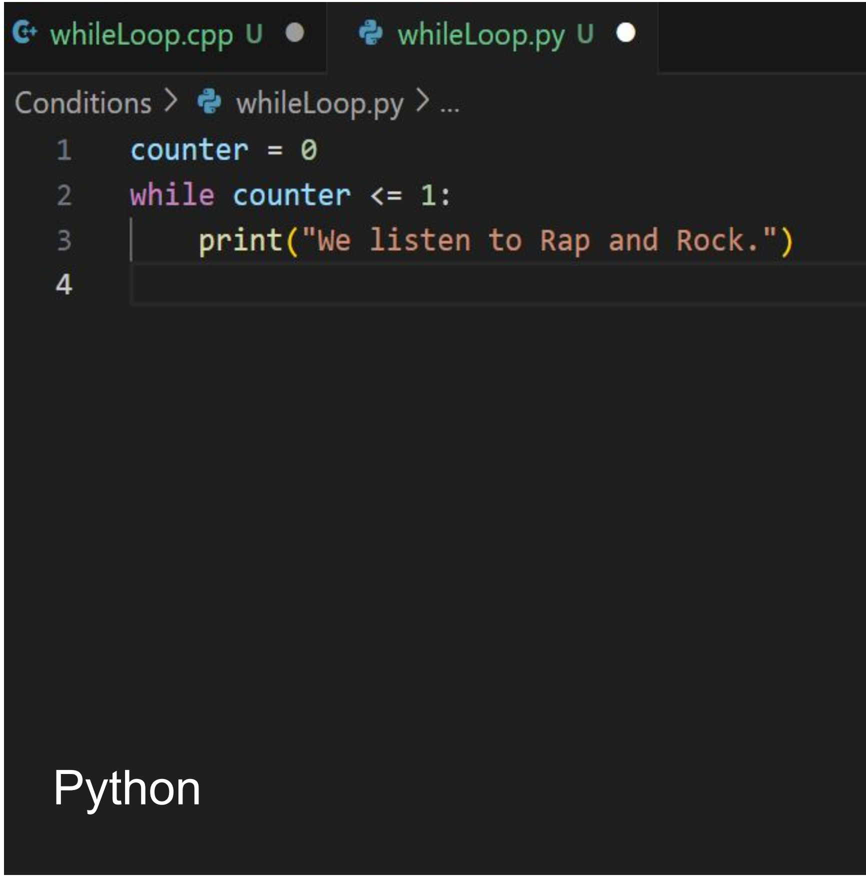
Python

While Loop in C++



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

int main(){
    int counter = 0;
    while (counter <= 1) {
        cout << "We listen to Rap and Rock." << endl;
    }
};
```

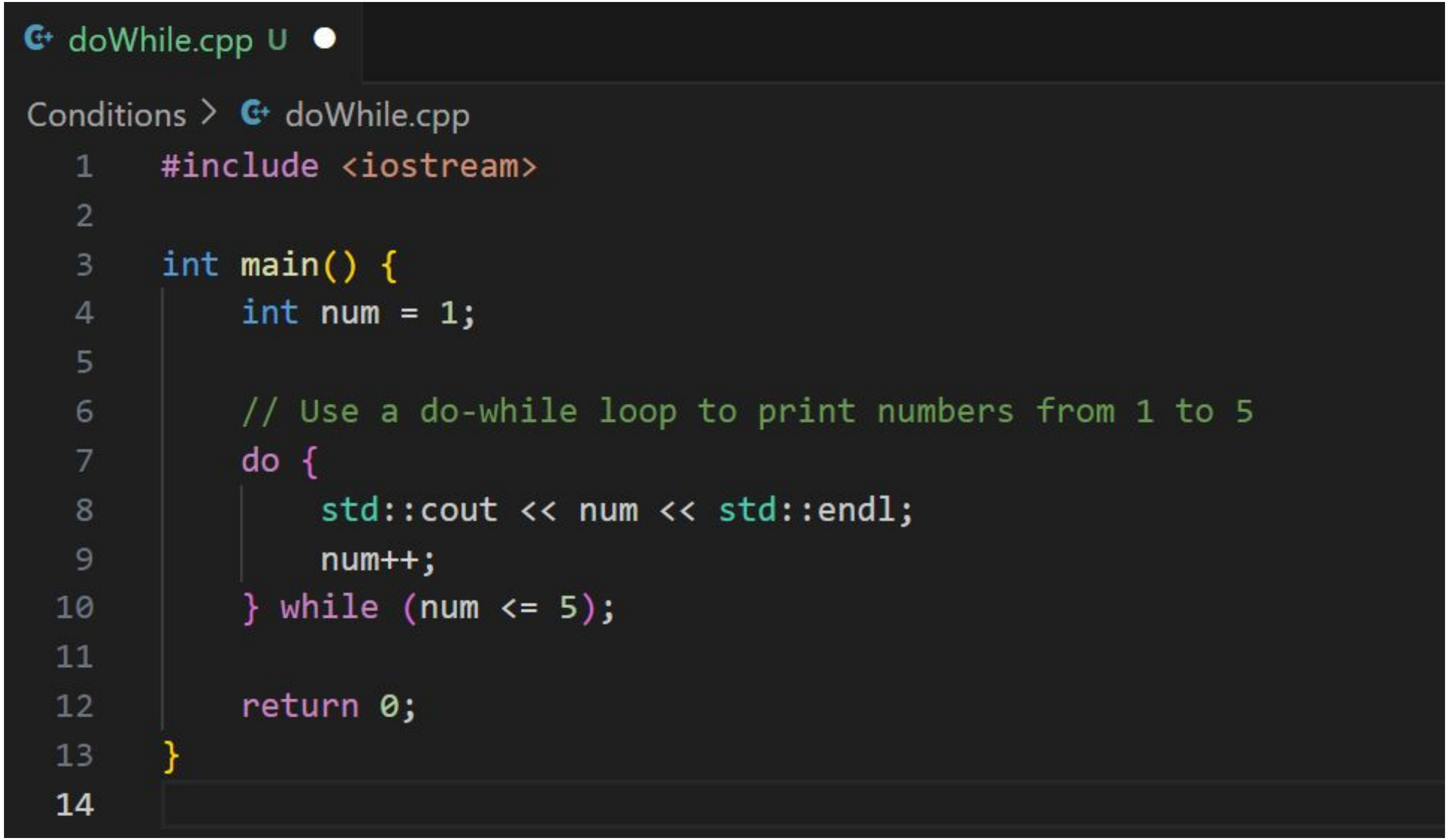
C++



```python
counter = 0
while counter <= 1:
    print("We listen to Rap and Rock.")

```

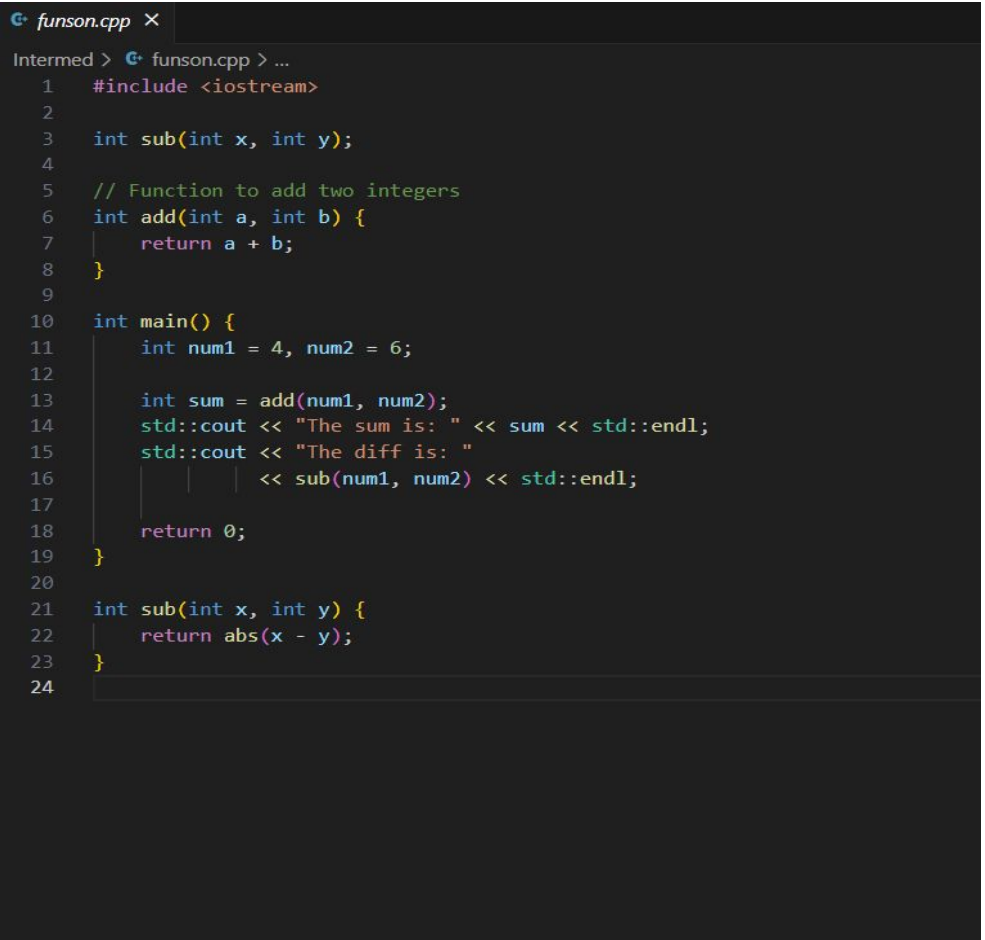
Python

Do While Loop in C++

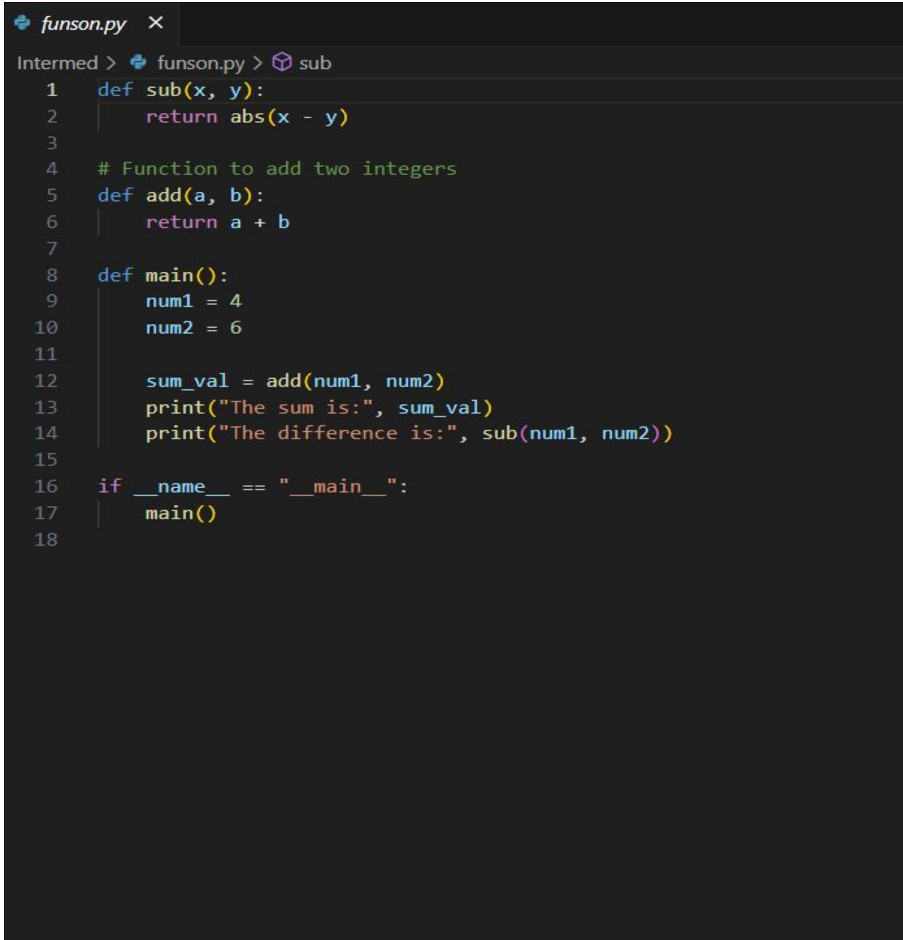


```cpp
#include <iostream>

int main() {
    int num = 1;

    // Use a do-while loop to print numbers from 1 to 5
    do {
        std::cout << num << std::endl;
        num++;
    } while (num <= 5);

    return 0;
}

```

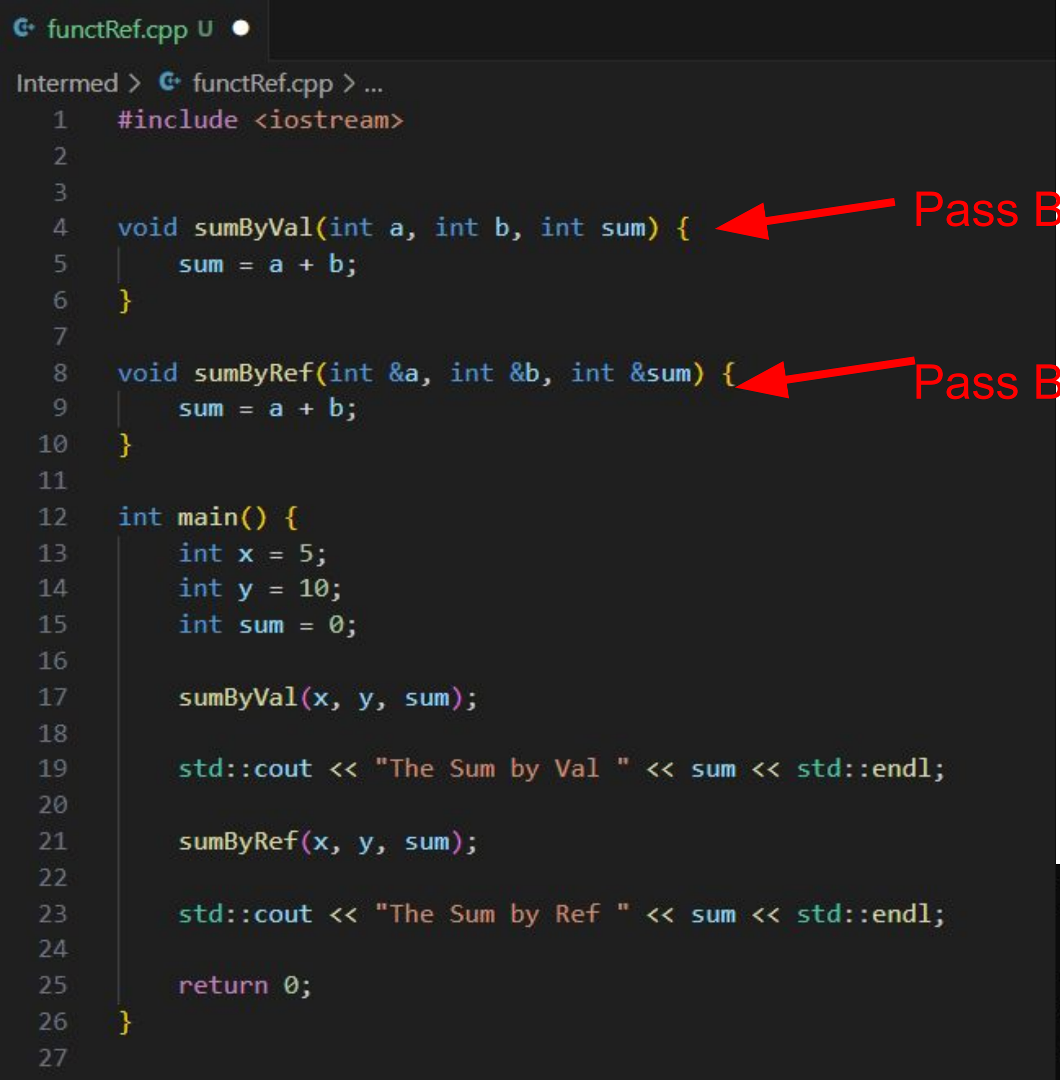
Functions



```cpp
#include <iostream>

int sub(int x, int y);

// Function to add two integers
int add(int a, int b) {
    return a + b;
}

int main() {
    int num1 = 4, num2 = 6;

    int sum = add(num1, num2);
    std::cout << "The sum is: " << sum << std::endl;
    std::cout << "The diff is: "
              << sub(num1, num2) << std::endl;

    return 0;
}

int sub(int x, int y) {
    return abs(x - y);
}

```



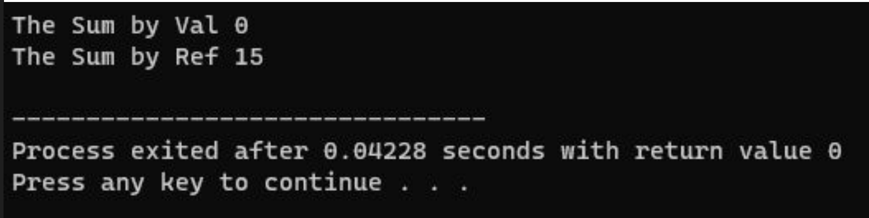
```python
def sub(x, y):
    return abs(x - y)

# Function to add two integers
def add(a, b):
    return a + b

def main():
    num1 = 4
    num2 = 6

    sum_val = add(num1, num2)
    print("The sum is:", sum_val)
    print("The difference is:", sub(num1, num2))

if __name__ == "__main__":
    main()

```



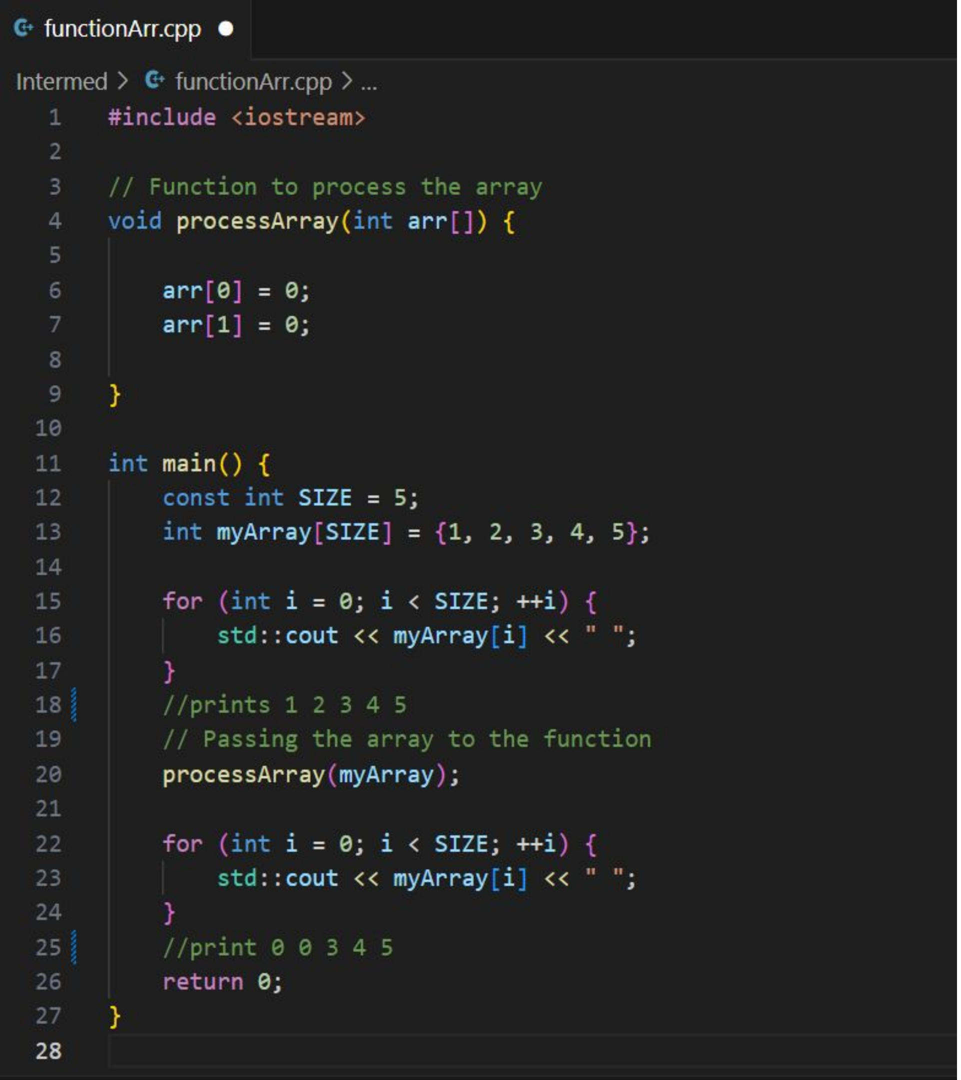
```cpp
#include <iostream>


void sumByVal(int a, int b, int sum) {
    sum = a + b;
}

void sumByRef(int &a, int &b, int &sum) {
    sum = a + b;
}

int main() {
    int x = 5;
    int y = 10;
    int sum = 0;

    sumByVal(x, y, sum);

    std::cout << "The Sum by Val " << sum << std::endl;

    sumByRef(x, y, sum);

    std::cout << "The Sum by Ref " << sum << std::endl;

    return 0;
}

```

Pass By Val

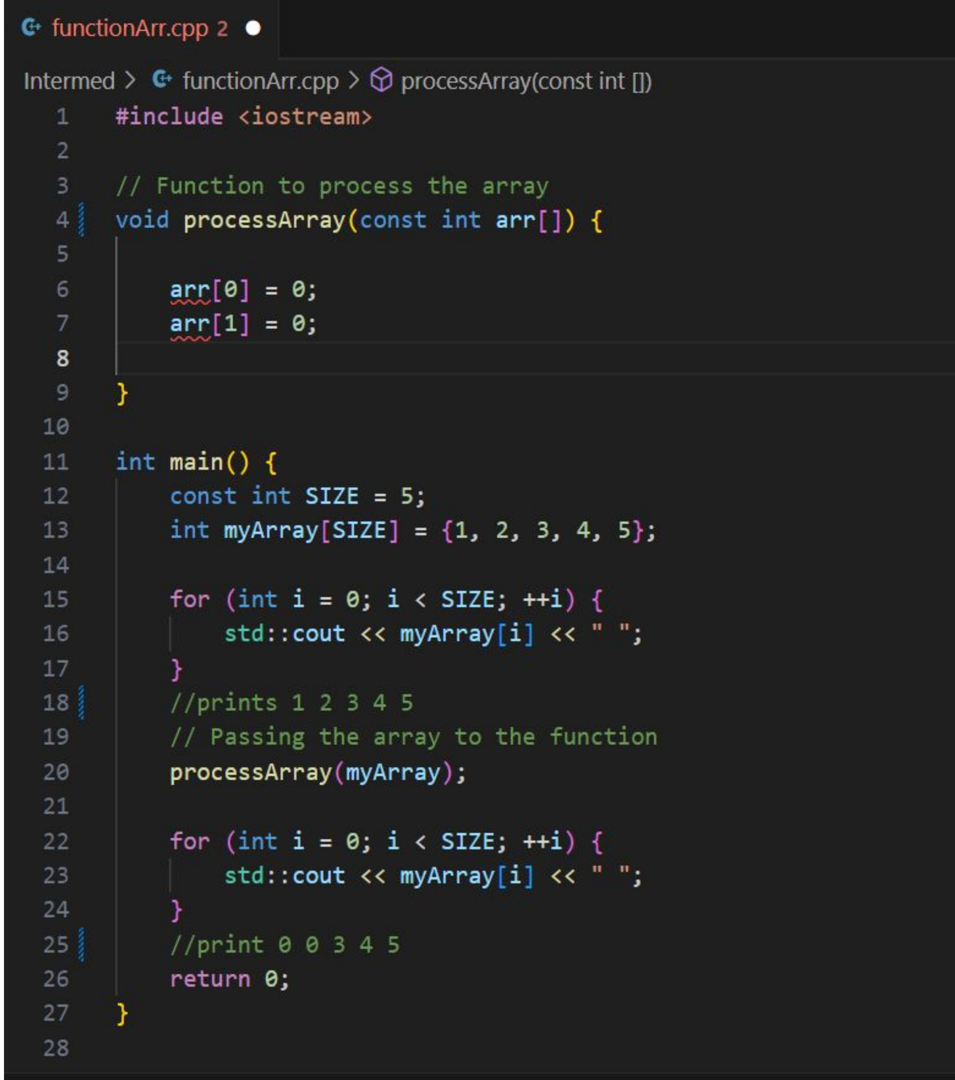
Pass By Ref



```
The Sum by Val 0
The Sum by Ref 15

--------------------------------
Process exited after 0.04228 seconds with return value 0
Press any key to continue . . .
```

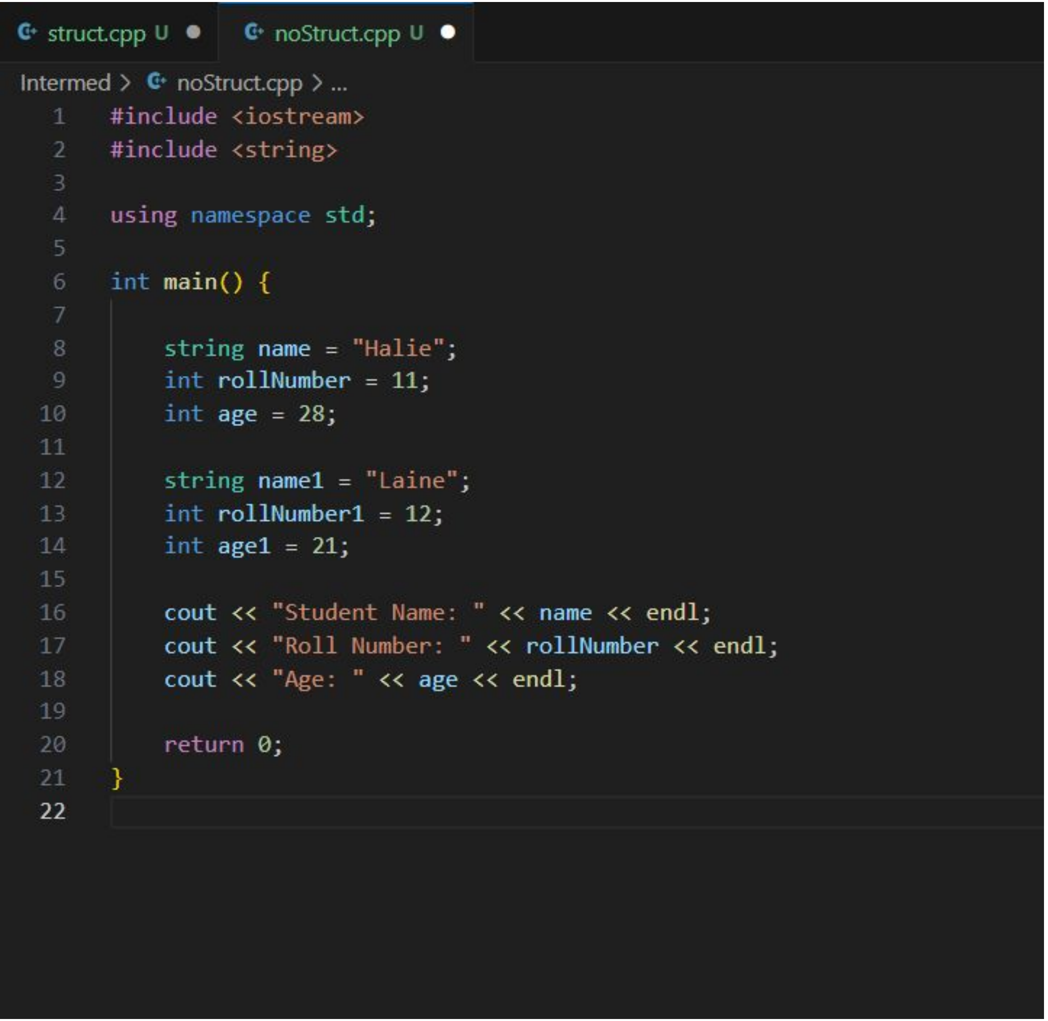


```cpp
#include <iostream>

// Function to process the array
void processArray(int arr[]) {

    arr[0] = 0;
    arr[1] = 0;

}

int main() {
    const int SIZE = 5;
    int myArray[SIZE] = {1, 2, 3, 4, 5};

    for (int i = 0; i < SIZE; ++i) {
        std::cout << myArray[i] << " ";
    }
    //prints 1 2 3 4 5
    // Passing the array to the function
    processArray(myArray);

    for (int i = 0; i < SIZE; ++i) {
        std::cout << myArray[i] << " ";
    }
    //print 0 0 3 4 5
    return 0;
}
```



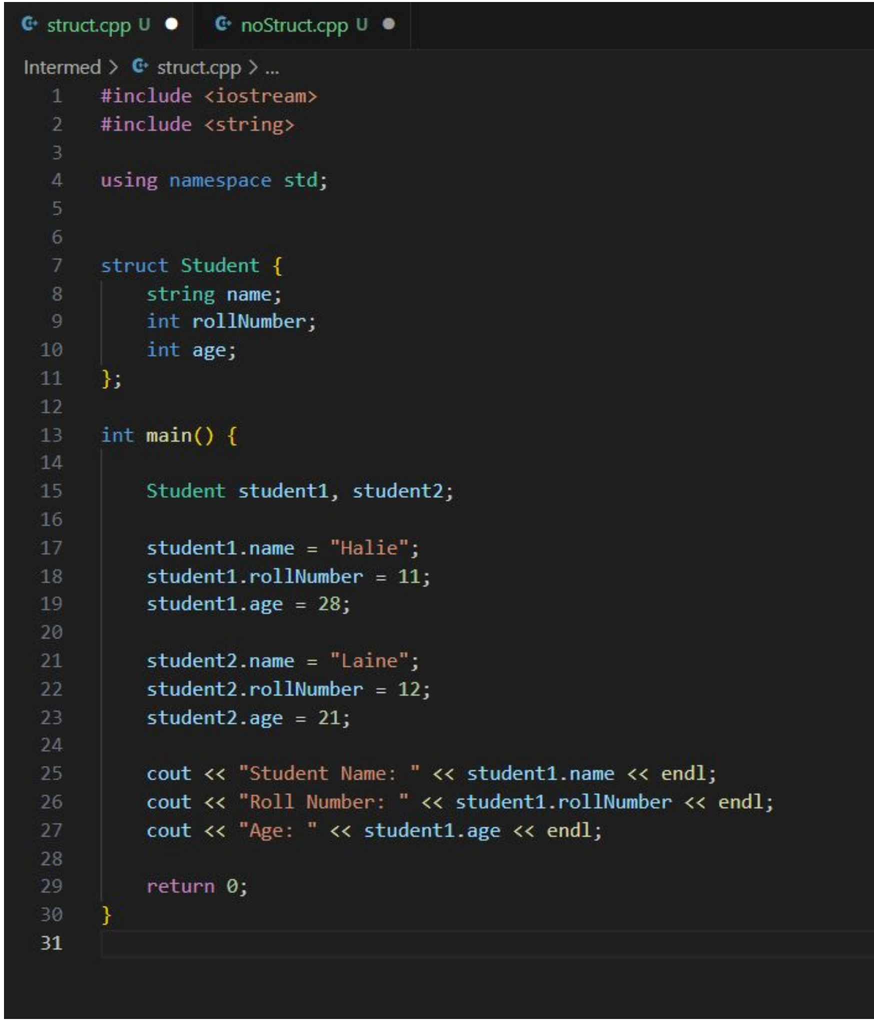
```cpp
#include <iostream>

// Function to process the array
void processArray(const int arr[]) {

    arr[0] = 0;
    arr[1] = 0;

}

int main() {
    const int SIZE = 5;
    int myArray[SIZE] = {1, 2, 3, 4, 5};

    for (int i = 0; i < SIZE; ++i) {
        std::cout << myArray[i] << " ";
    }
    //prints 1 2 3 4 5
    // Passing the array to the function
    processArray(myArray);

    for (int i = 0; i < SIZE; ++i) {
        std::cout << myArray[i] << " ";
    }
    //print 0 0 3 4 5
    return 0;
}
```

Without struct



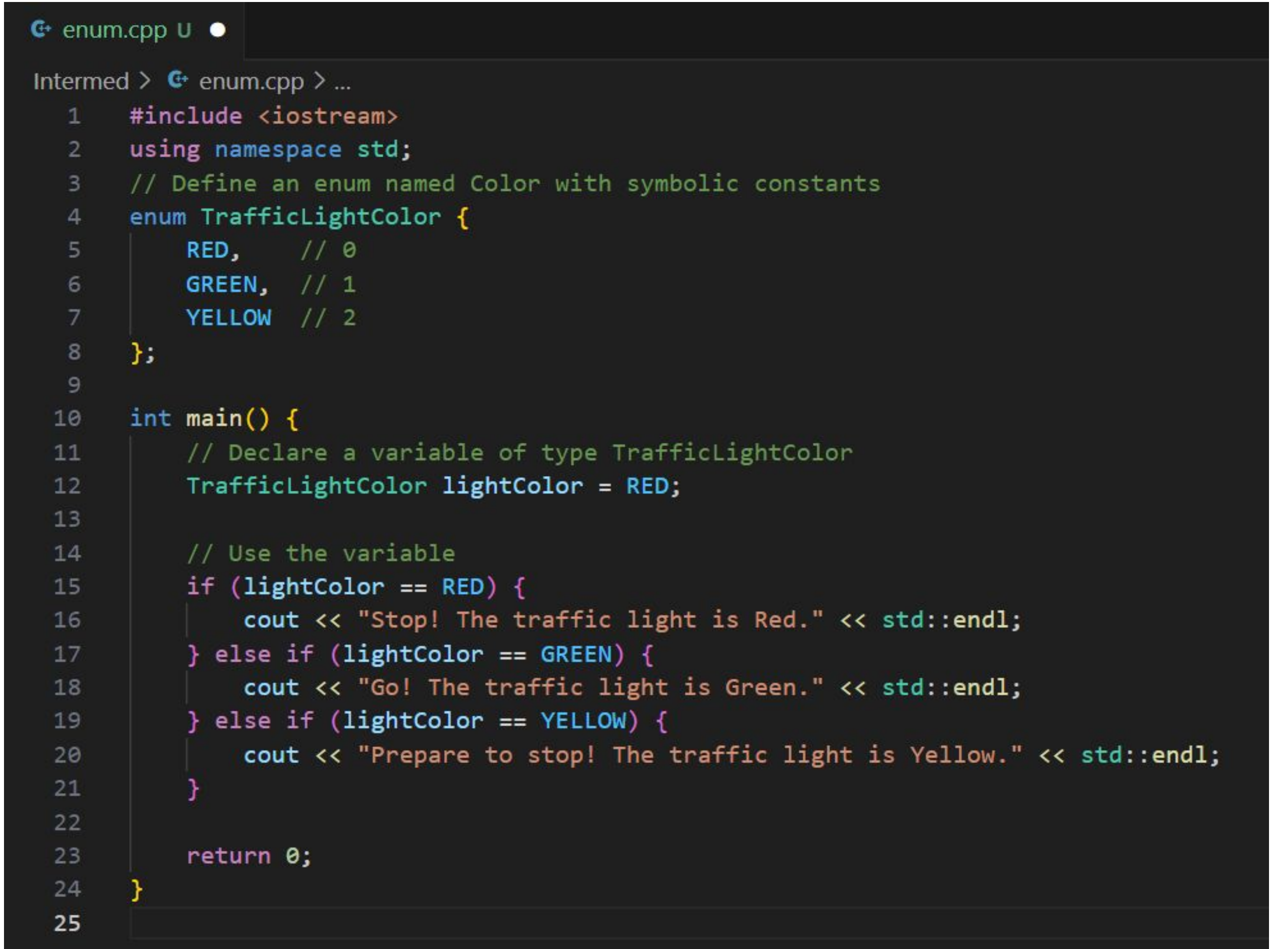
```cpp
#include <iostream>
#include <string>

using namespace std;

int main() {

    string name = "Halie";
    int rollNumber = 11;
    int age = 28;

    string name1 = "Laine";
    int rollNumber1 = 12;
    int age1 = 21;

    cout << "Student Name: " << name << endl;
    cout << "Roll Number: " << rollNumber << endl;
    cout << "Age: " << age << endl;

    return 0;
}

```

With struct



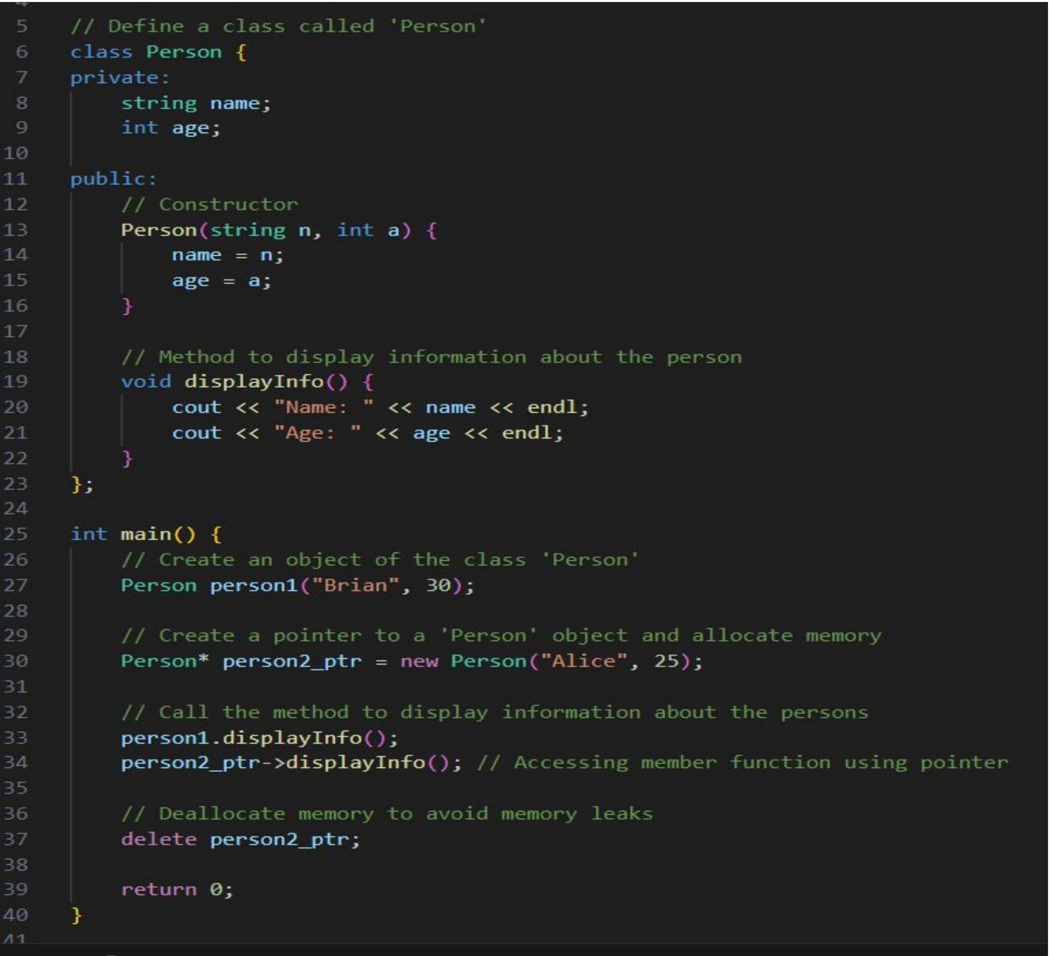
```cpp
#include <iostream>
#include <string>

using namespace std;


struct Student {
    string name;
    int rollNumber;
    int age;
};

int main() {

    Student student1, student2;

    student1.name = "Halie";
    student1.rollNumber = 11;
    student1.age = 28;

    student2.name = "Laine";
    student2.rollNumber = 12;
    student2.age = 21;

    cout << "Student Name: " << student1.name << endl;
    cout << "Roll Number: " << student1.rollNumber << endl;
    cout << "Age: " << student1.age << endl;

    return 0;
}

```

Enum

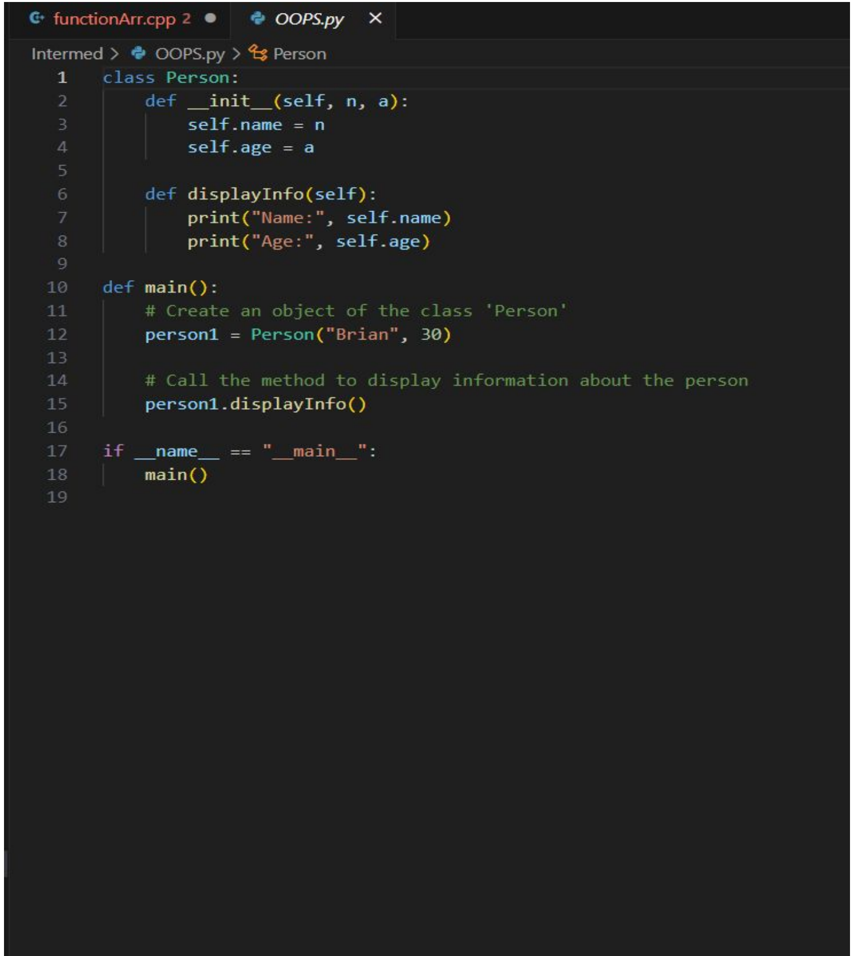


```cpp
#include <iostream>
using namespace std;
// Define an enum named Color with symbolic constants
enum TrafficLightColor {
    RED,    // 0
    GREEN,  // 1
    YELLOW  // 2
};

int main() {
    // Declare a variable of type TrafficLightColor
    TrafficLightColor lightColor = RED;

    // Use the variable
    if (lightColor == RED) {
        cout << "Stop! The traffic light is Red." << std::endl;
    } else if (lightColor == GREEN) {
        cout << "Go! The traffic light is Green." << std::endl;
    } else if (lightColor == YELLOW) {
        cout << "Prepare to stop! The traffic light is Yellow." << std::endl;
    }

    return 0;
}

```

Class



```cpp
// Define a class called 'Person'
class Person {
private:
    string name;
    int age;

public:
    // Constructor
    Person(string n, int a) {
        name = n;
        age = a;
    }

    // Method to display information about the person
    void displayInfo() {
        cout << "Name: " << name << endl;
        cout << "Age: " << age << endl;
    }
};

int main() {
    // Create an object of the class 'Person'
    Person person1("Brian", 30);

    // Create a pointer to a 'Person' object and allocate memory
    Person* person2_ptr = new Person("Alice", 25);

    // Call the method to display information about the persons
    person1.displayInfo();
    person2_ptr->displayInfo(); // Accessing member function using pointer

    // Deallocate memory to avoid memory leaks
    delete person2_ptr;

    return 0;
}
```



```python
class Person:
    def __init__(self, n, a):
        self.name = n
        self.age = a

    def displayInfo(self):
        print("Name:", self.name)
        print("Age:", self.age)

def main():
    # Create an object of the class 'Person'
    person1 = Person("Brian", 30)

    # Call the method to display information about the person
    person1.displayInfo()

if __name__ == "__main__":
    main()
```

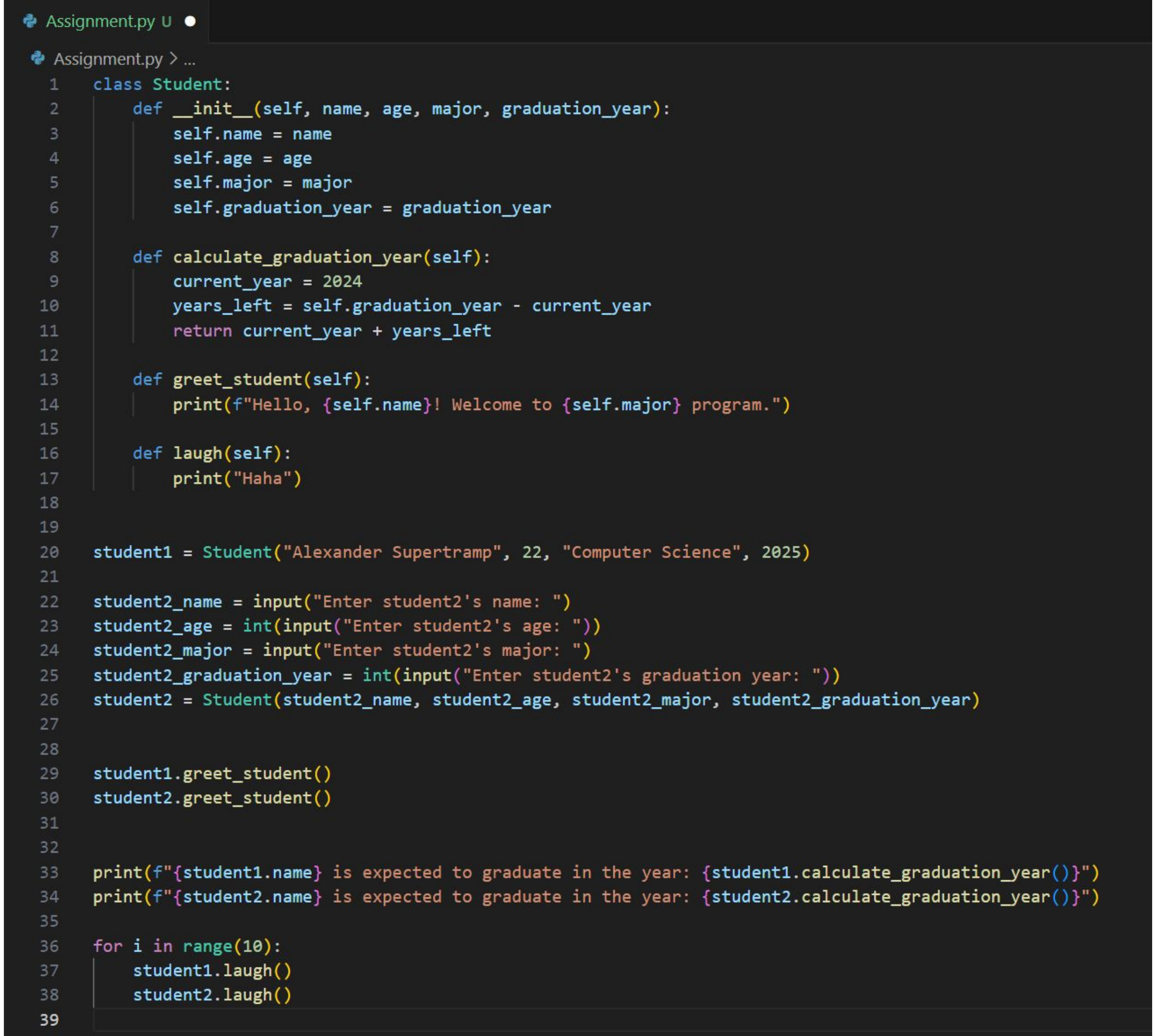
Assignment on next slide

Convert the Python code to C++.

You are free to work in a group of two if you want, just write both of your name as a multiple line comment on the top.
But, if you work in a group, then you must write comment on your code

Do not ask AI (chatgpt, gemini, copilot, grok or other) to convert your code. Although, you may ask for some smaller task like how to do this.

Slides are always the most powerful source for you to do assignment.



```python
class Student:
    def __init__(self, name, age, major, graduation_year):
        self.name = name
        self.age = age
        self.major = major
        self.graduation_year = graduation_year

    def calculate_graduation_year(self):
        current_year = 2024
        years_left = self.graduation_year - current_year
        return current_year + years_left

    def greet_student(self):
        print(f"Hello, {self.name}! Welcome to {self.major} program.")

    def laugh(self):
        print("Haha")


student1 = Student("Alexander Supertramp", 22, "Computer Science", 2025)

student2_name = input("Enter student2's name: ")
student2_age = int(input("Enter student2's age: "))
student2_major = input("Enter student2's major: ")
student2_graduation_year = int(input("Enter student2's graduation year: "))
student2 = Student(student2_name, student2_age, student2_major, student2_graduation_year)


student1.greet_student()
student2.greet_student()


print(f"{student1.name} is expected to graduate in the year: {student1.calculate_graduation_year()}")
print(f"{student2.name} is expected to graduate in the year: {student2.calculate_graduation_year()}")

for i in range(10):
    student1.laugh()
    student2.laugh()

```

C++ - OneCompiler - Write, run and share C++ code online

Control + click (win)
Command + click (mac)

After you complete,
Try running it and
Click triple dot next to
Run button and
download

Url to code:
https://github.com/Khagendra01/Cpp_In_One_Hour/blob/main/Assignment.py

Thank You!

https://www.linkedin.com/in/khagendrakhatri/

https://github.com/khagendra01/

K-Gen

Khagendra Khatri