GETTING STARTED WITH C++ PROGRAMMING

INTRODUCTION TO C++ PROGRAMMING

INTRODUCTION

❑ C Vs. C++ Language

C Language

- Extension: .c
- For output/input
  - **printf**, **scanf** with placeholder
- Variable type:
  - `int, float, char, double`
  - `char [ ], bool`

C++ Language

- Extension: .cpp
- For output/input
  - **cout**, **cin** without placeholder
- Variable type:
  - `int, float, char, double`
  - `char [ ], bool,` **`string`**

INTRODUCTION

❑ Sample code: C Vs. C++

C syntax

```
#include<stdio.h>

main(){

        char name[20];

        printf(“Hello world!\n”);

        printf(“What is your name?: ”);

        scanf(“%s”, &name);

        printf(“Hi, %s \n”, name);

        printf(“Bye!”);

}
```

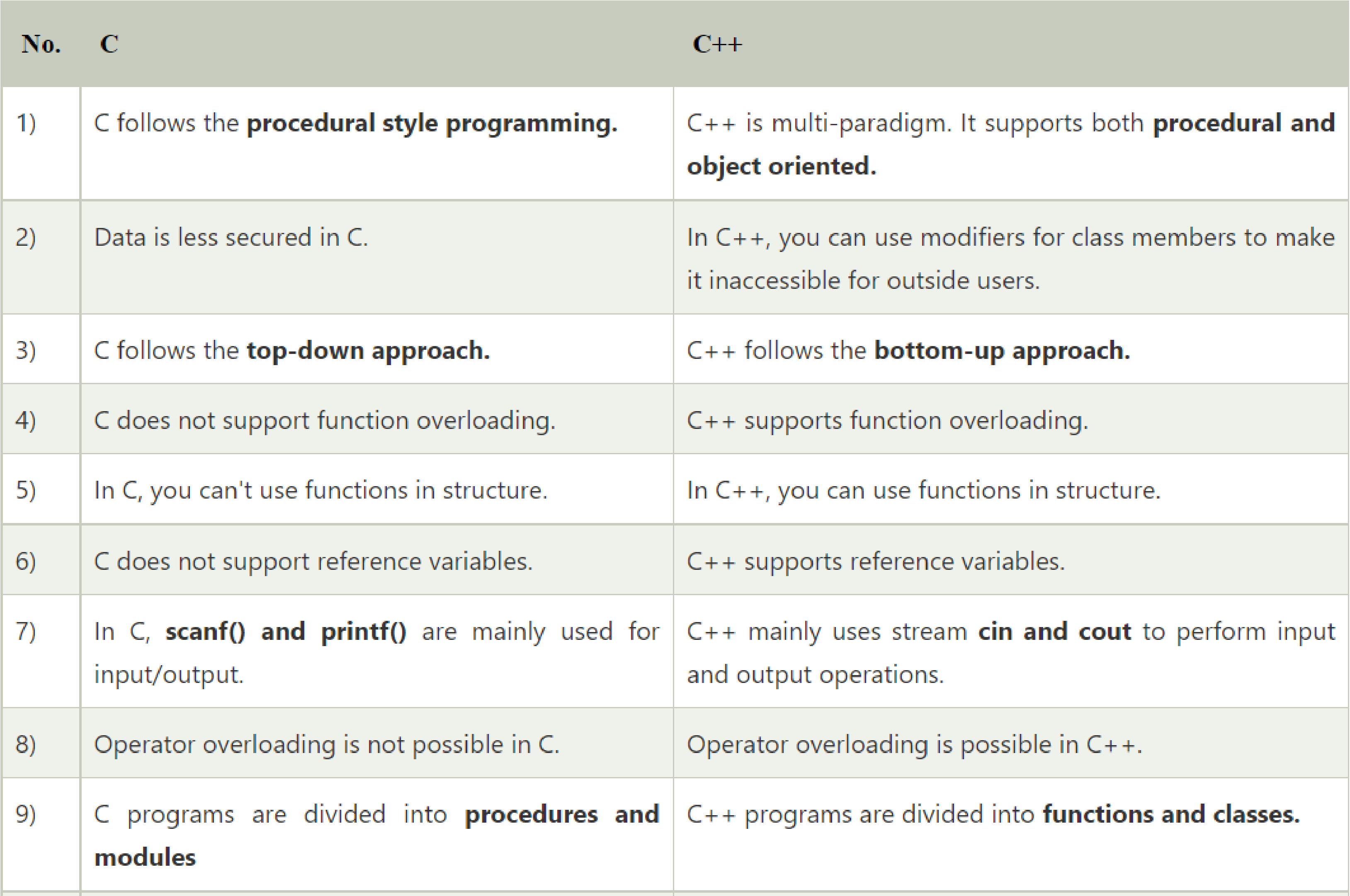
Filename: test.c

C++ syntax

```
#include<iostream>

using namespace std;


main(){

        string name;

        cout<<“Hello world!\n”;

        cout<<“What is your name?: ”;

        cin>>name;

        cout<<“Hi, ”<<name<<endl;

        cout<<“Bye!”;


}
```

Filename: test.cpp

MORE ON C AND C++



No.	C	C++
1)	C follows the **procedural style programming.**	C++ is multi-paradigm. It supports both **procedural and object oriented.**
2)	Data is less secured in C.	In C++, you can use modifiers for class members to make it inaccessible for outside users.
3)	C follows the **top-down approach.**	C++ follows the **bottom-up approach.**
4)	C does not support function overloading.	C++ supports function overloading.
5)	In C, you can't use functions in structure.	In C++, you can use functions in structure.
6)	C does not support reference variables.	C++ supports reference variables.
7)	In C, **scanf() and printf()** are mainly used for input/output.	C++ mainly uses stream **cin and cout** to perform input and output operations.
8)	Operator overloading is not possible in C.	Operator overloading is possible in C++.
9)	C programs are divided into **procedures and modules**	C++ programs are divided into **functions and classes.**

https://www.geeksforgeeks.org/difference-between-c-and-c/

SYNTAX

❑ Most C and C++ syntax are the same

- Decision making
  - `if, else if, else`
- Loop
  - `for, while, do while`
- Function

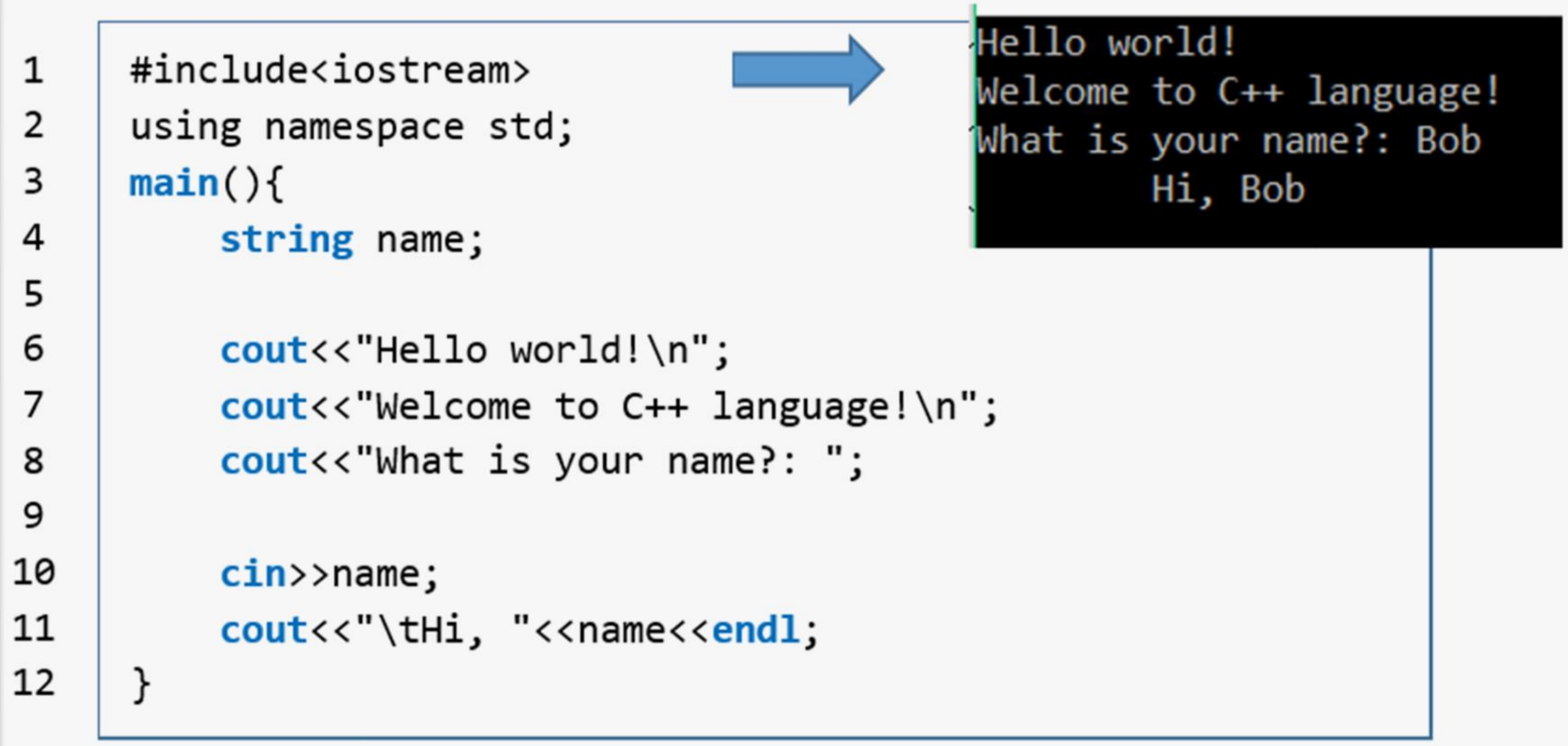
C and C++ are the same

- **Structure**
  - Just in C++, after the creation of structure, we don't need to use **struct** keyword again for creating variable
    - `struct Student{ ... };`
    - `Student st;`

Examples

REVIEW

- **Function**
- **Structure**
- **Loop**
- **Other**

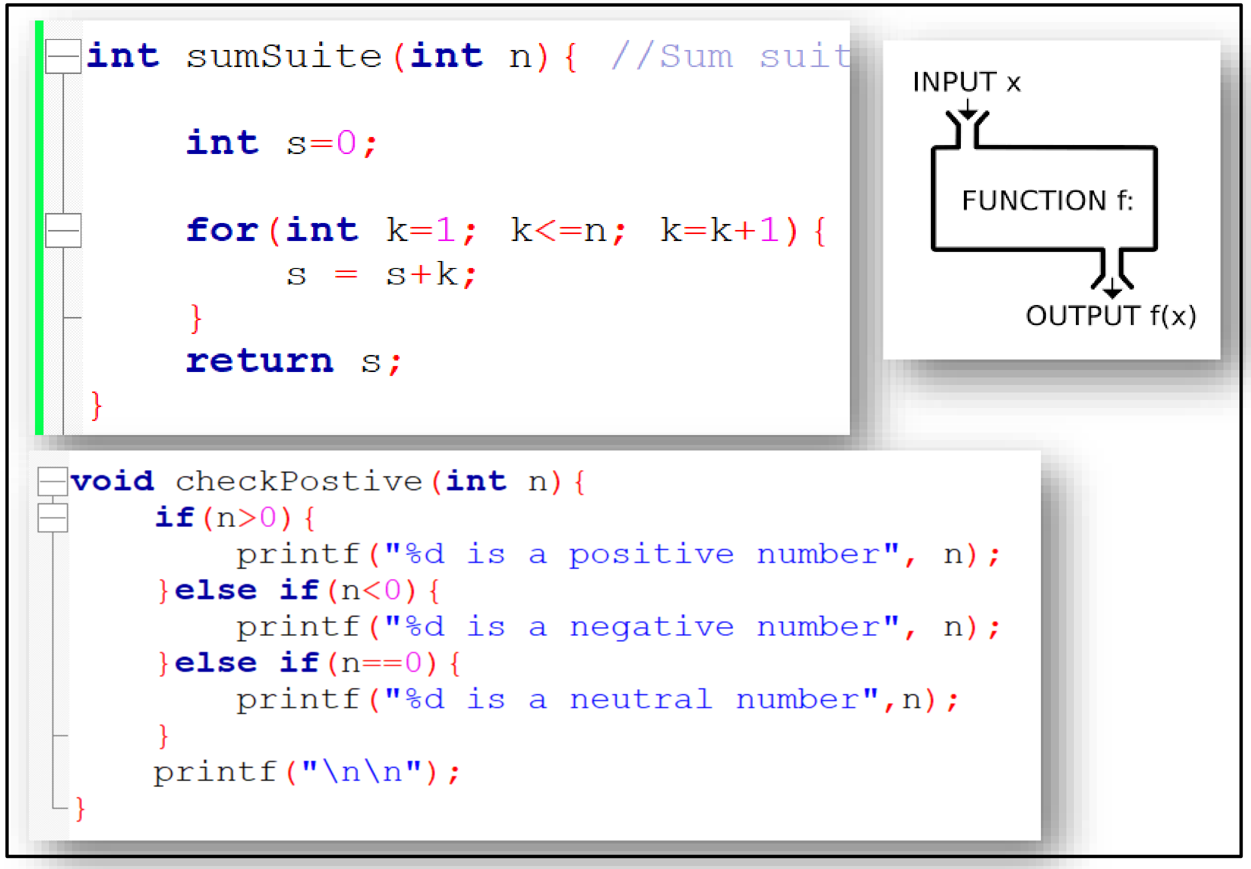


```cpp
#include<iostream>
using namespace std;
main(){
    string name;

    cout<<"Hello world!\n";
    cout<<"Welcome to C++ language!\n";
    cout<<"What is your name?: ";

    cin>>name;
    cout<<"\tHi, "<<name<<endl;
}
```

```
Hello world!
Welcome to C++ language!
What is your name?: Bob
        Hi, Bob
```

Figure 1: A basic C++ program overview

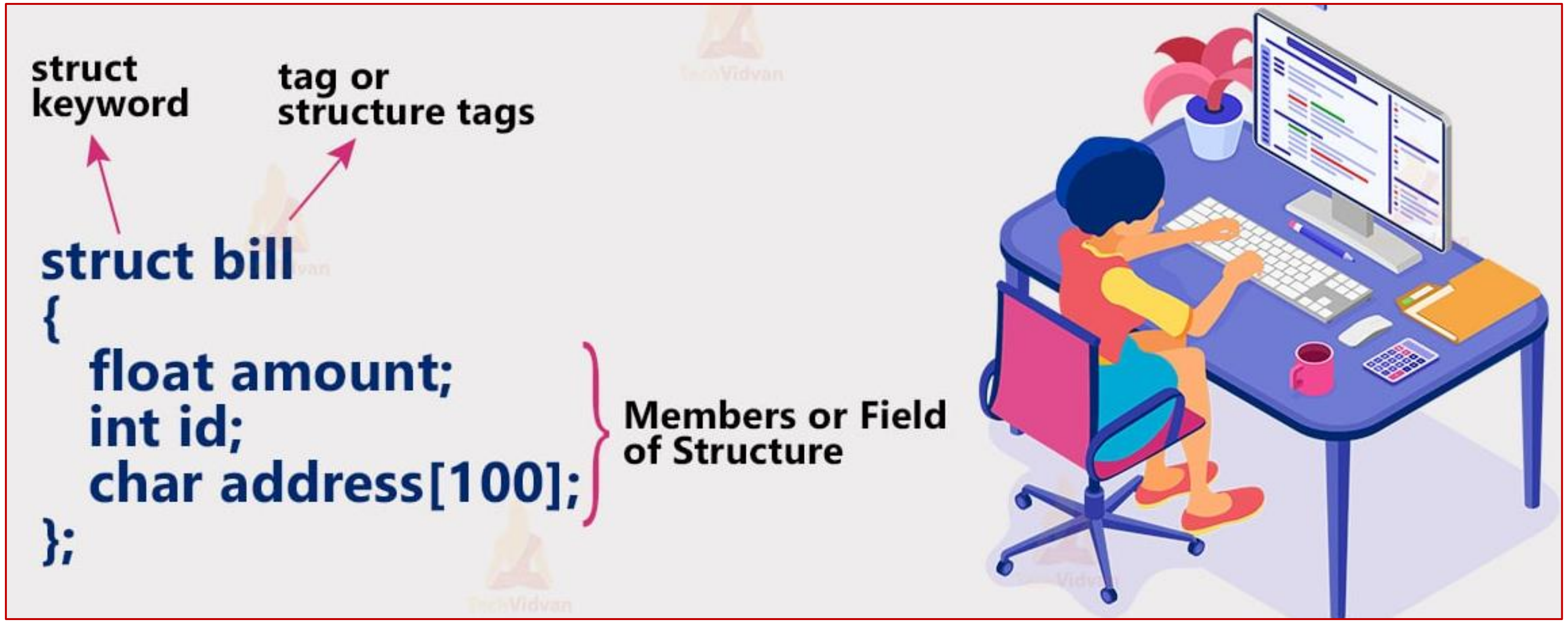
FUNCTION



```c
int sumSuite(int n){ //Sum suit

    int s=0;

    for(int k=1; k<=n; k=k+1){
        s = s+k;
    }
    return s;
}
```

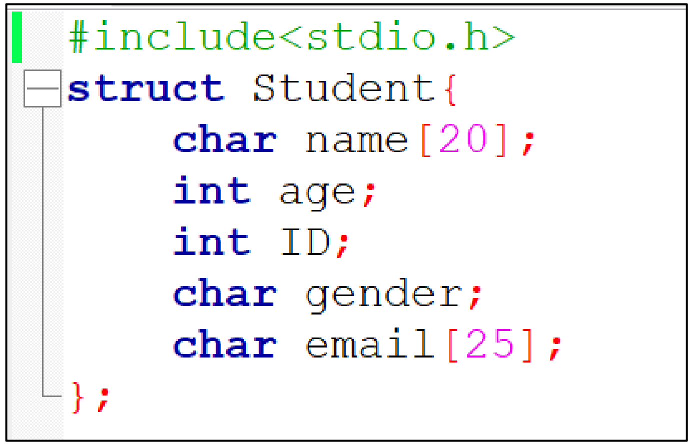
```c
void checkPostive(int n){
    if(n>0){
        printf("%d is a positive number", n);
    }else if(n<0){
        printf("%d is a negative number", n);
    }else if(n==0){
        printf("%d is a neutral number",n);
    }
    printf("\n\n");
}
```

FUNCTION



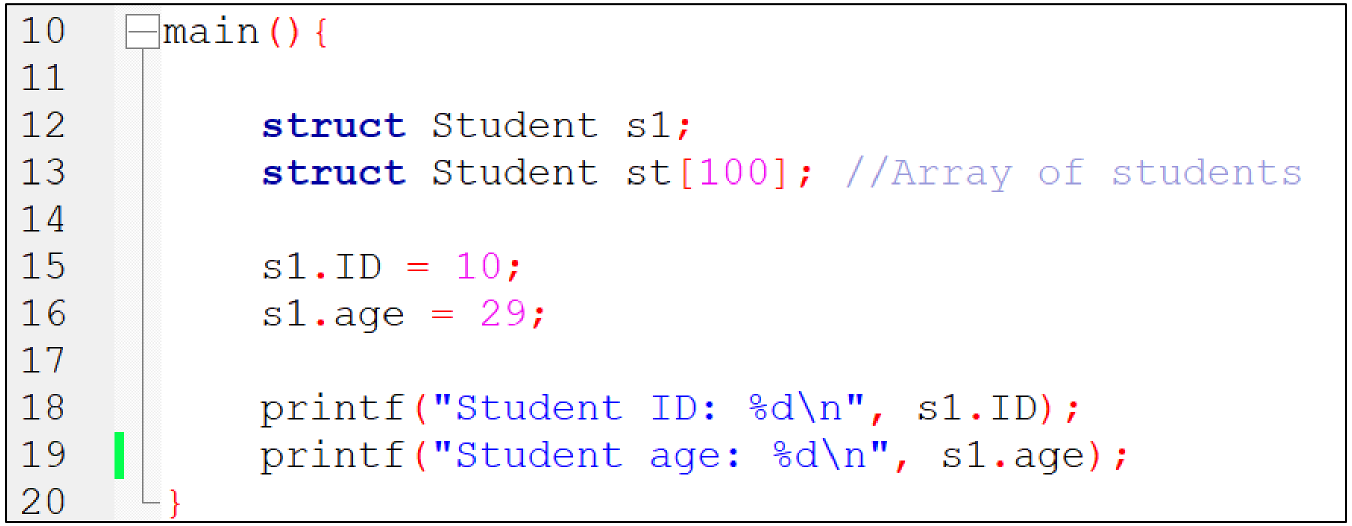
```
struct bill
{
   float amount;
   int id;
   char address[100];
};
```

STRUCTURE



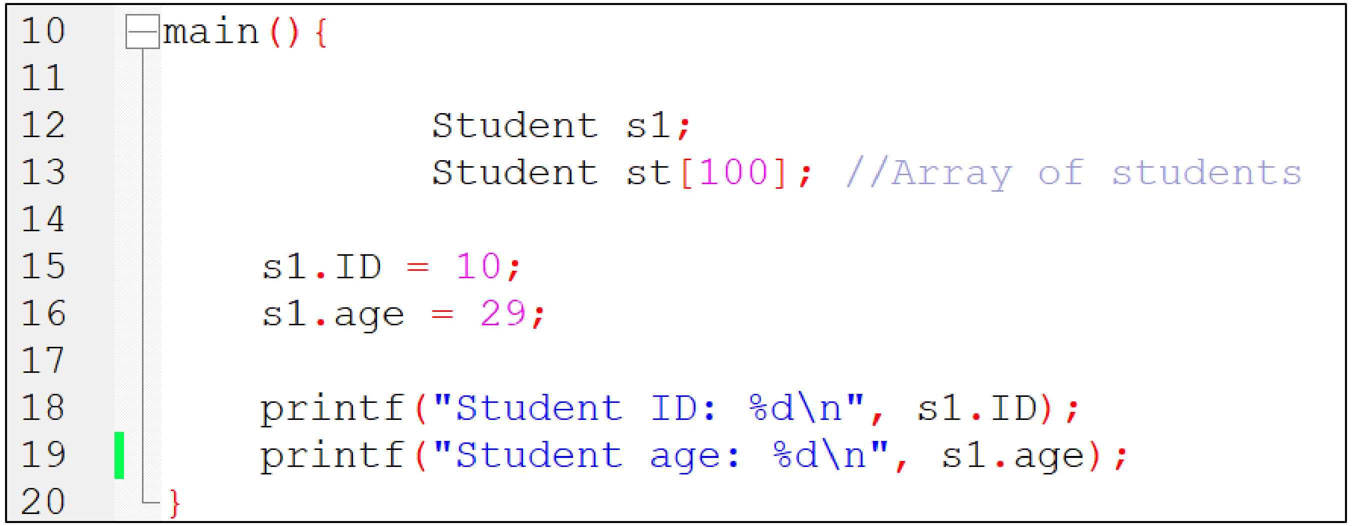
```c
#include<stdio.h>
struct Student{
    char name[20];
    int age;
    int ID;
    char gender;
    char email[25];
};
```

C program



```c
main(){

    struct Student s1;
    struct Student st[100]; //Array of students

    s1.ID = 10;
    s1.age = 29;

    printf("Student ID: %d\n", s1.ID);
    printf("Student age: %d\n", s1.age);
}
```

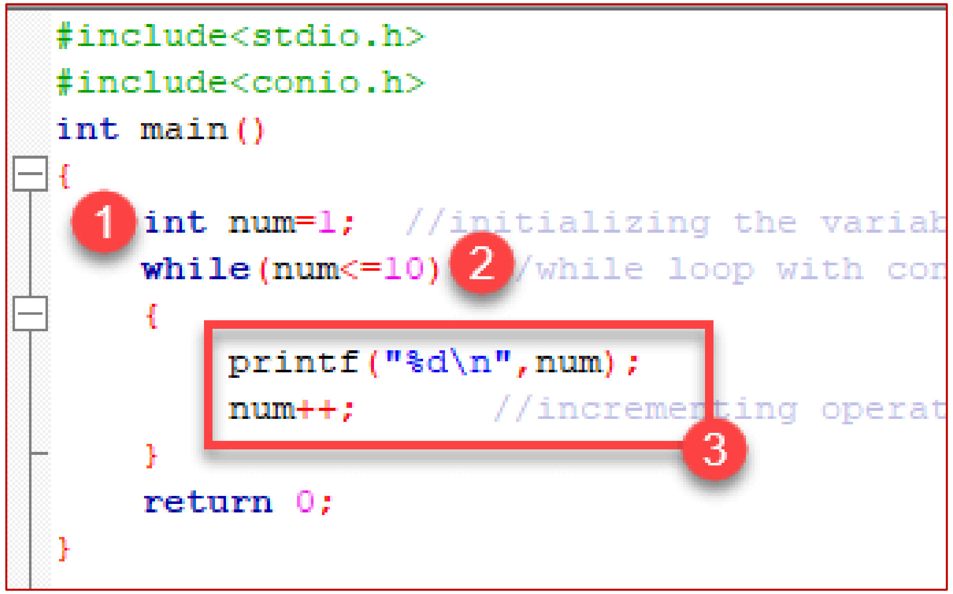
C++ program



```cpp
main(){

           Student s1;
           Student st[100]; //Array of students

    s1.ID = 10;
    s1.age = 29;

    printf("Student ID: %d\n", s1.ID);
    printf("Student age: %d\n", s1.age);
}
```

LOOP

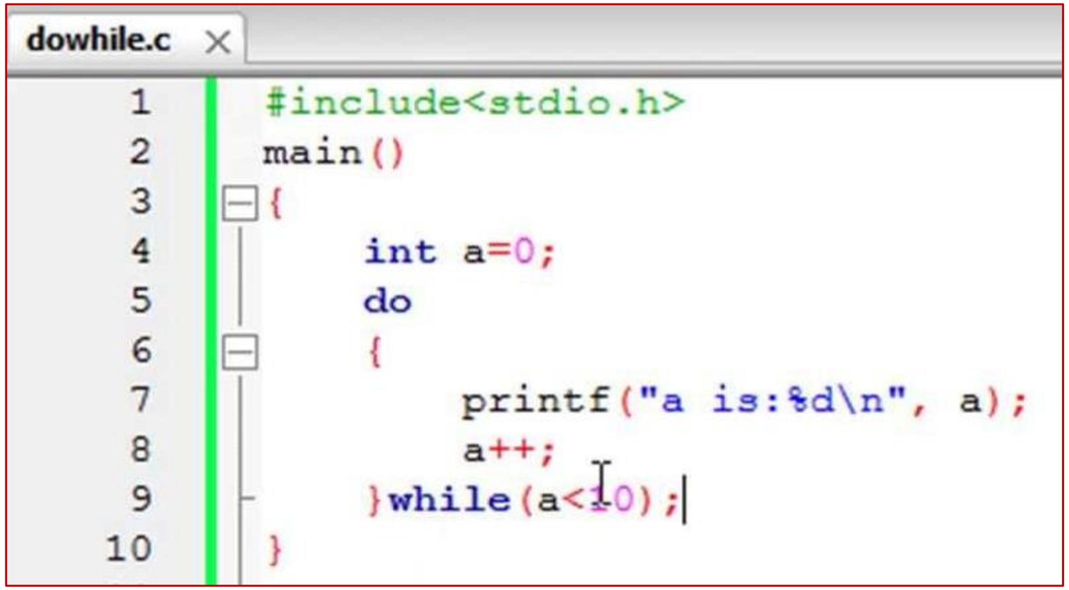
What is the output from the following program?



```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num=1;    //initializing the variab
    while(num<=10)   /while loop with con
    {
        printf("%d\n",num);
        num++;        //incrementing operat
    }
    return 0;
}
```

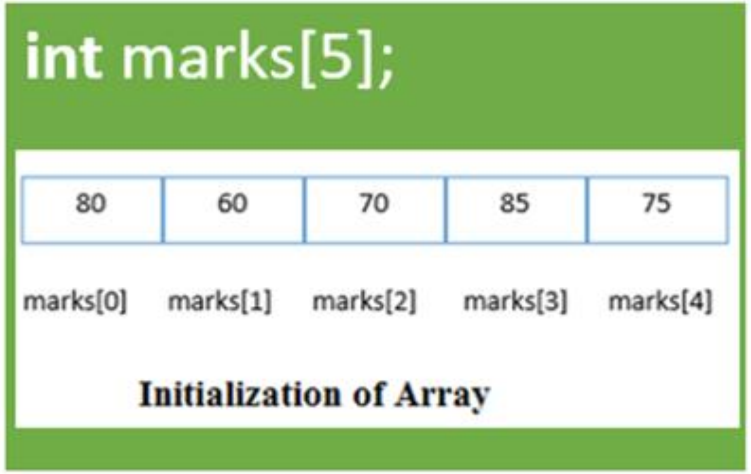
LOOP

What is the output from the following program?



```c
#include<stdio.h>
main()
{
    int a=0;
    do
    {
        printf("a is:%d\n", a);
        a++;
    }while(a<10);
}
```

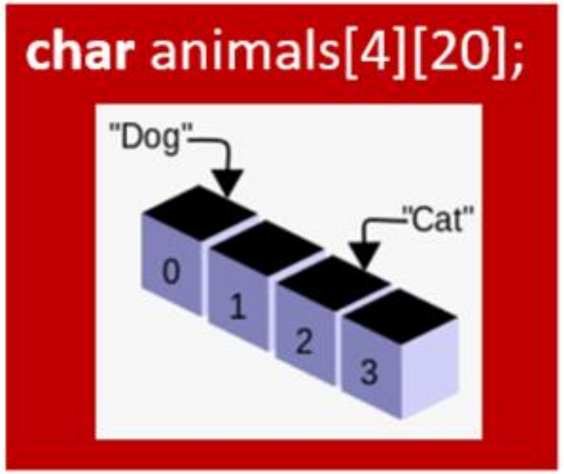
ARRAY



```
int marks[5];
```

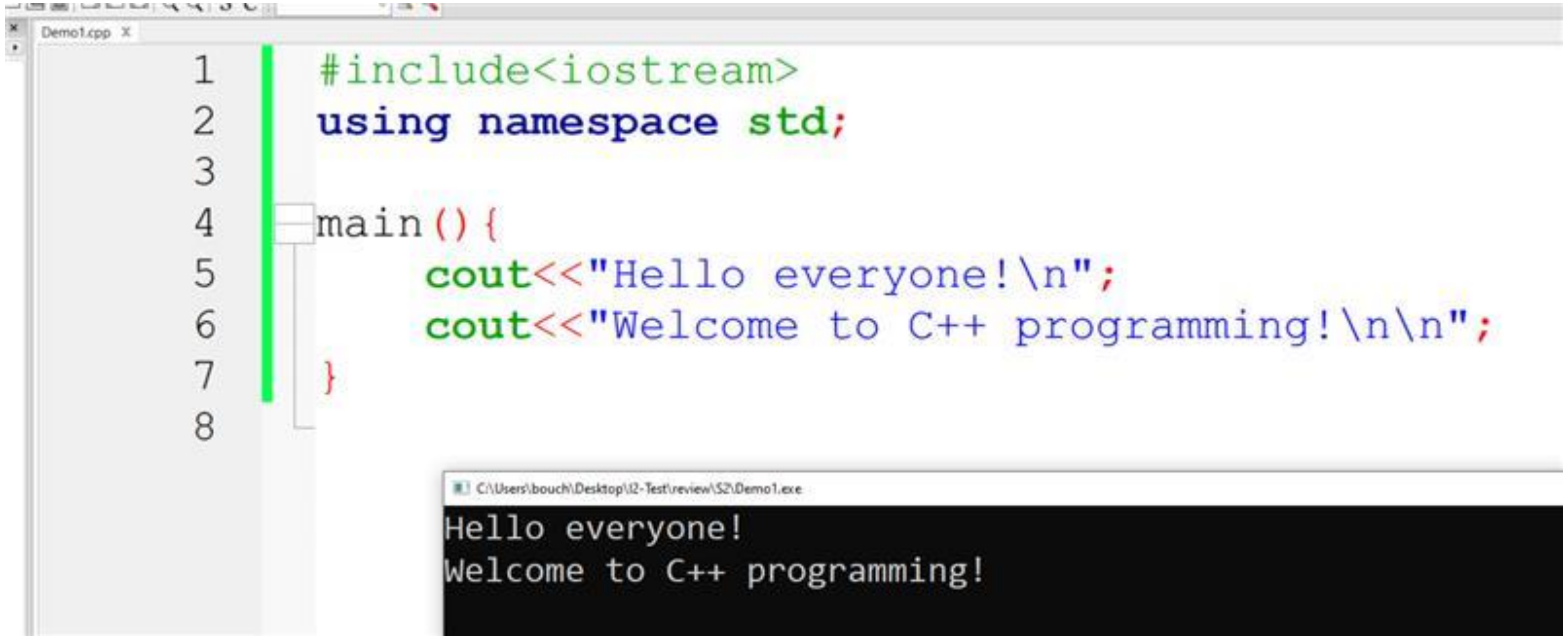
80	60	70	85	75
marks[0]	marks[1]	marks[2]	marks[3]	marks[4]

Initialization of Array



```
char animals[4][20];
```

EXAMPLES: C++ PROGRAMMING

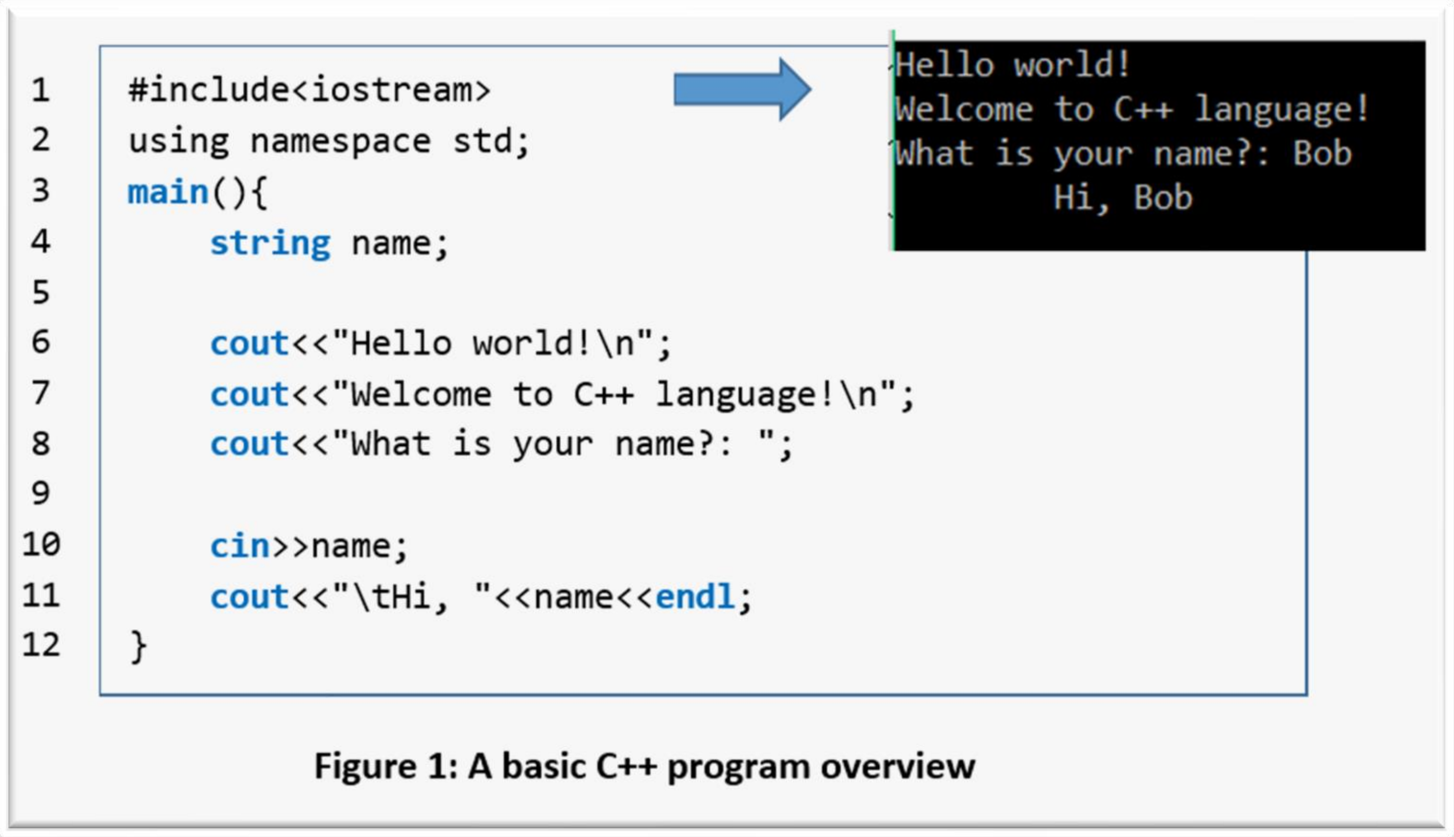


```cpp
#include<iostream>
using namespace std;

main(){
	cout<<"Hello everyone!\n";
	cout<<"Welcome to C++ programming!\n\n";
}

```

```
Hello everyone!
Welcome to C++ programming!
```

1st program in C++ programming

EXAMPLES: C++ PROGRAMMING



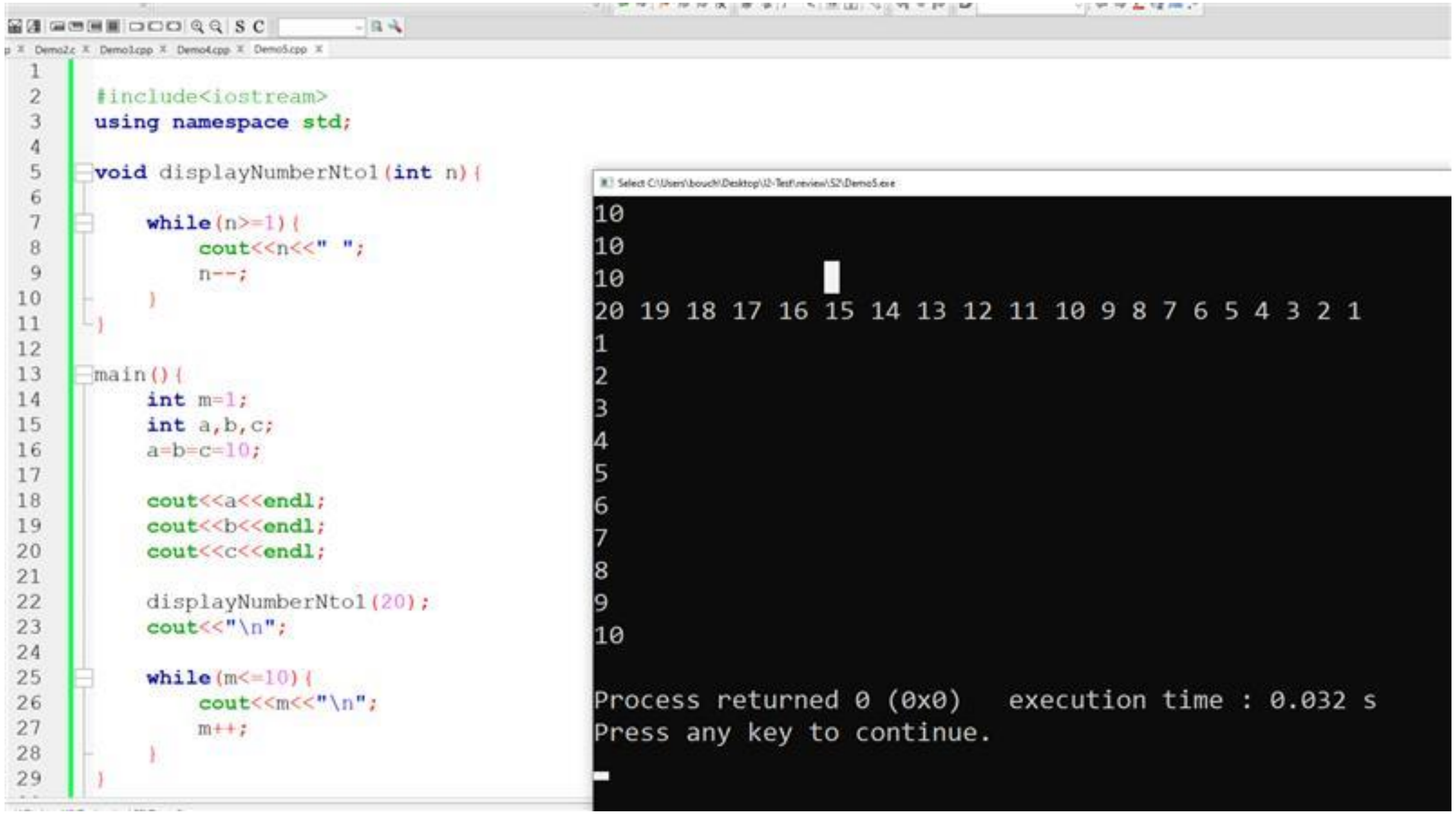
```cpp
#include<iostream>
using namespace std;
main(){
    string name;

    cout<<"Hello world!\n";
    cout<<"Welcome to C++ language!\n";
    cout<<"What is your name?: ";

    cin>>name;
    cout<<"\tHi, "<<name<<endl;
}
```

```
Hello world!
Welcome to C++ language!
What is your name?: Bob
        Hi, Bob
```

Figure 1: A basic C++ program overview

Using **cin** and **cout** to get input and display

EXAMPLES: C++ PROGRAMMING

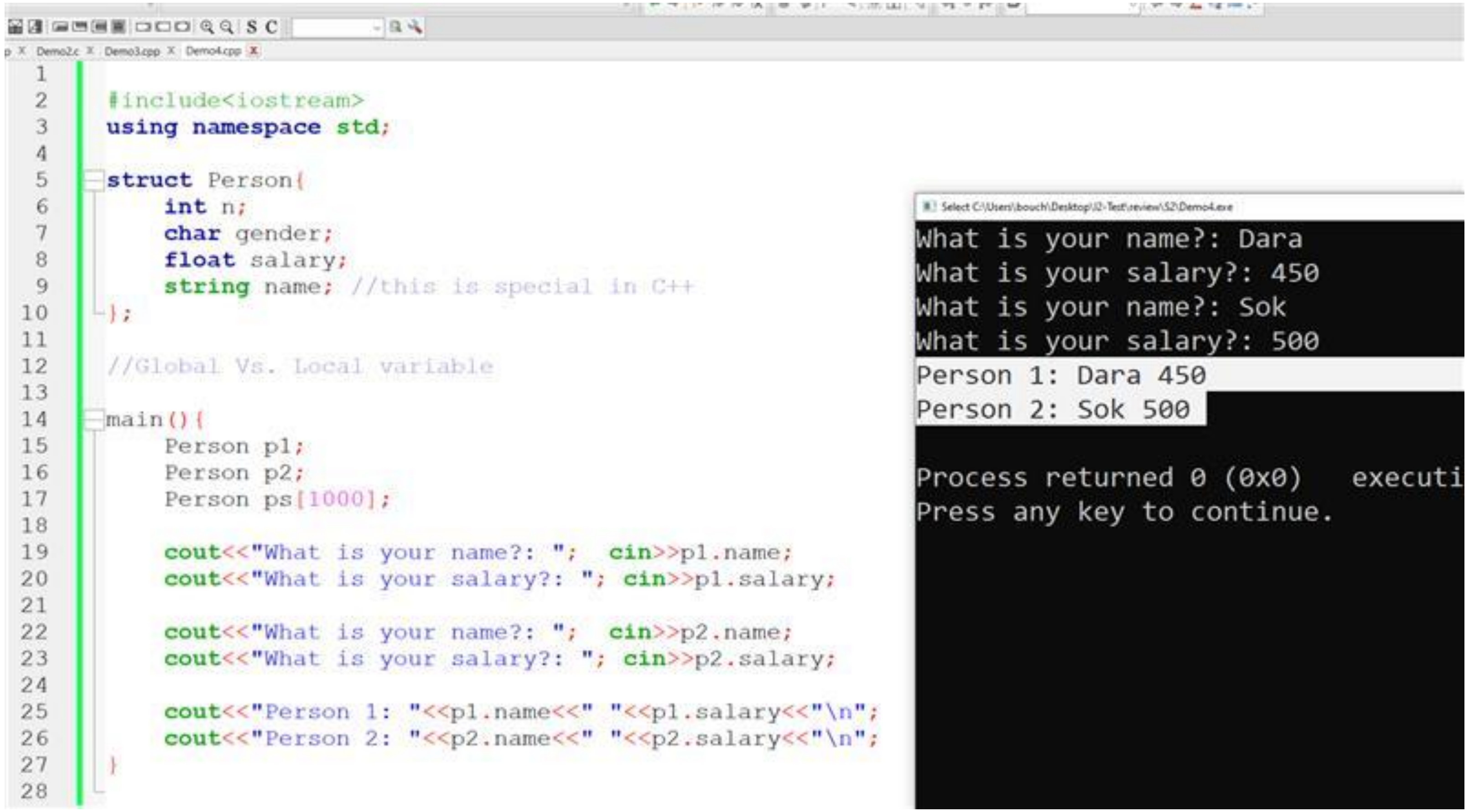


```cpp

#include<iostream>
using namespace std;

void displayNumberNto1(int n){

    while(n>=1){
        cout<<n<<" ";
        n--;
    }
}

main(){
    int m=1;
    int a,b,c;
    a=b=c=10;

    cout<<a<<endl;
    cout<<b<<endl;
    cout<<c<<endl;

    displayNumberNto1(20);
    cout<<"\n";

    while(m<=10){
        cout<<m<<"\n";
        m++;
    }
}
```

```
10
10
10
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
1
2
3
4
5
6
7
8
9
10

Process returned 0 (0x0)   execution time : 0.032 s
Press any key to continue.
```

Using function and loop in C++

EXAMPLES: C++ PROGRAMMING

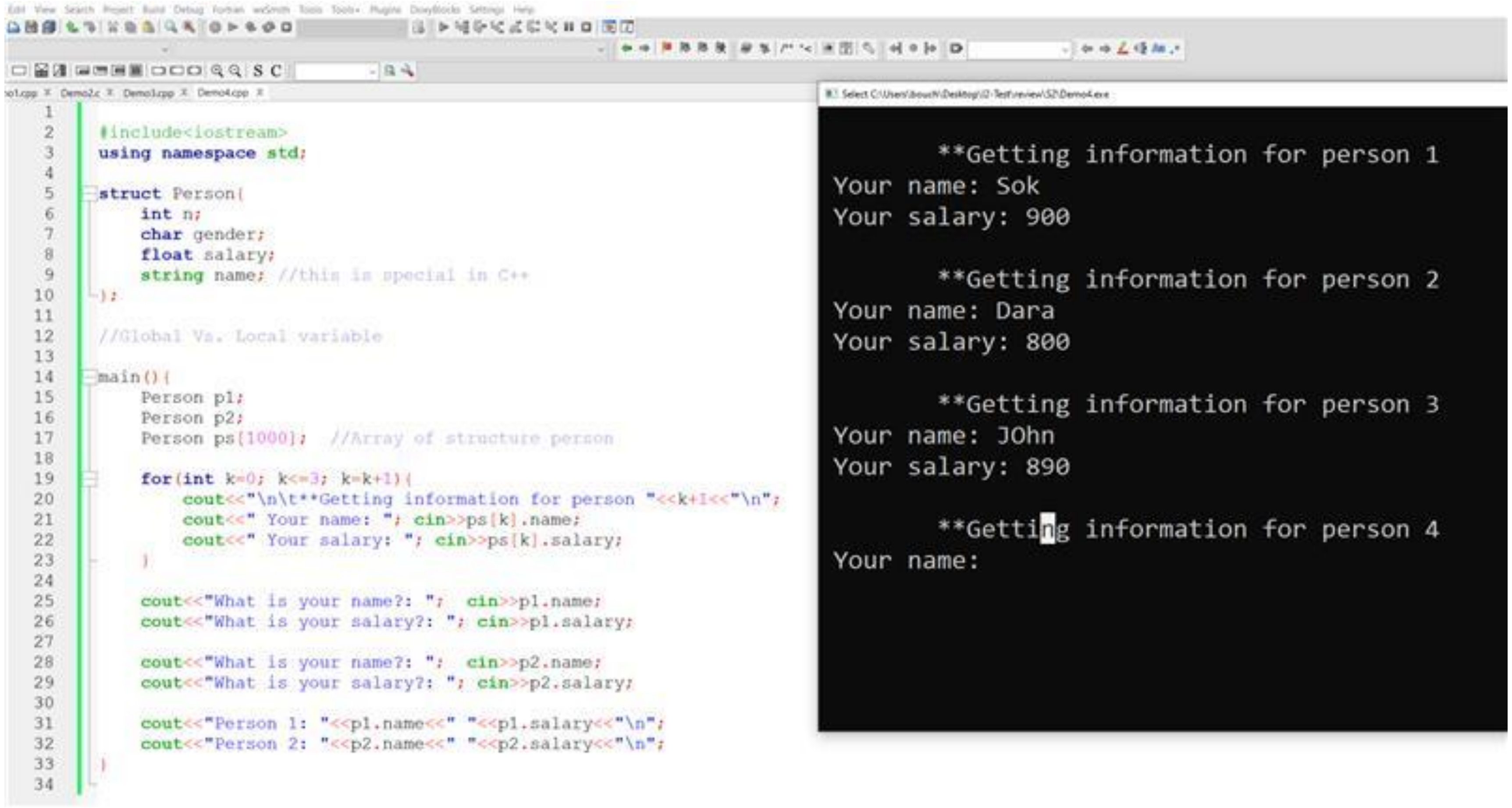


```cpp

#include<iostream>
using namespace std;

struct Person{
    int n;
    char gender;
    float salary;
    string name; //this is special in C++
};

//Global Vs. Local variable

main(){
    Person p1;
    Person p2;
    Person ps[1000];

    cout<<"What is your name?: ";  cin>>p1.name;
    cout<<"What is your salary?: "; cin>>p1.salary;

    cout<<"What is your name?: ";  cin>>p2.name;
    cout<<"What is your salary?: "; cin>>p2.salary;

    cout<<"Person 1: "<<p1.name<<" "<<p1.salary<<"\n";
    cout<<"Person 2: "<<p2.name<<" "<<p2.salary<<"\n";
}

```

```
What is your name?: Dara
What is your salary?: 450
What is your name?: Sok
What is your salary?: 500
Person 1: Dara 450
Person 2: Sok 500

Process returned 0 (0x0)   executi
Press any key to continue.
```

Using structure in C++

EXAMPLES: C++ PROGRAMMING



```cpp

#include<iostream>
using namespace std;

struct Person{
	int n;
	char gender;
	float salary;
	string name; //this is special in C++
};

//Global Vs. Local variable

main(){
	Person p1;
	Person p2;
	Person ps[1000];  //Array of structure person

	for(int k=0; k<=3; k=k+1){
		cout<<"\n\t**Getting information for person "<<k+1<<"\n";
		cout<<" Your name: "; cin>>ps[k].name;
		cout<<" Your salary: "; cin>>ps[k].salary;
	}

	cout<<"What is your name?: ";  cin>>p1.name;
	cout<<"What is your salary?: "; cin>>p1.salary;

	cout<<"What is your name?: ";  cin>>p2.name;
	cout<<"What is your salary?: "; cin>>p2.salary;

	cout<<"Person 1: "<<p1.name<<" "<<p1.salary<<"\n";
	cout<<"Person 2: "<<p2.name<<" "<<p2.salary<<"\n";
}

```

```

        **Getting information for person 1
 Your name: Sok
 Your salary: 900

        **Getting information for person 2
 Your name: Dara
 Your salary: 800

        **Getting information for person 3
 Your name: JOhn
 Your salary: 890

        **Getting information for person 4
 Your name:
```

Using array of structure in C++