



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++

```
#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!");
```

```
}
```

C
syntax

Filename: test.c

```
#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!";
```

```
}
```

C++
syntax

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 .

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

```
1  #include<iostream>
2  using namespace std;
3  main(){
4      string name;
5
6      cout<<"Hello world!\n";
7      cout<<"Welcome to C++ language!\n";
8      cout<<"What is your name?: ";
9
10     cin>>name;
11     cout<<"\tHi, "<<name<<endl;
12 }
```



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

Figure 1: A basic C++ program overview

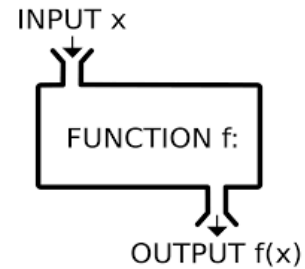
FUNCTION

```
int sumSuite(int n) { //Sum suite

    int s=0;

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

```
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 keyword

tag or structure tags

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

Members or Field of Structure



The illustration shows a person with dark hair, wearing a red shirt and blue shorts, sitting on a pink office chair at a blue desk. They are using a computer with a large monitor displaying a web application. On the desk, there is a keyboard, a mouse, a pink mug, and a calculator. A potted plant with red flowers is on the desk. The background is a light gray wall with a faint 'TechVidvan' logo.

STRUCTURE

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

C program

```
10 main() {
11
12     struct Student s1;
13     struct Student st[100]; //Array of students
14
15     s1.ID = 10;
16     s1.age = 29;
17
18     printf("Student ID: %d\n", s1.ID);
19     printf("Student age: %d\n", s1.age);
20 }
```

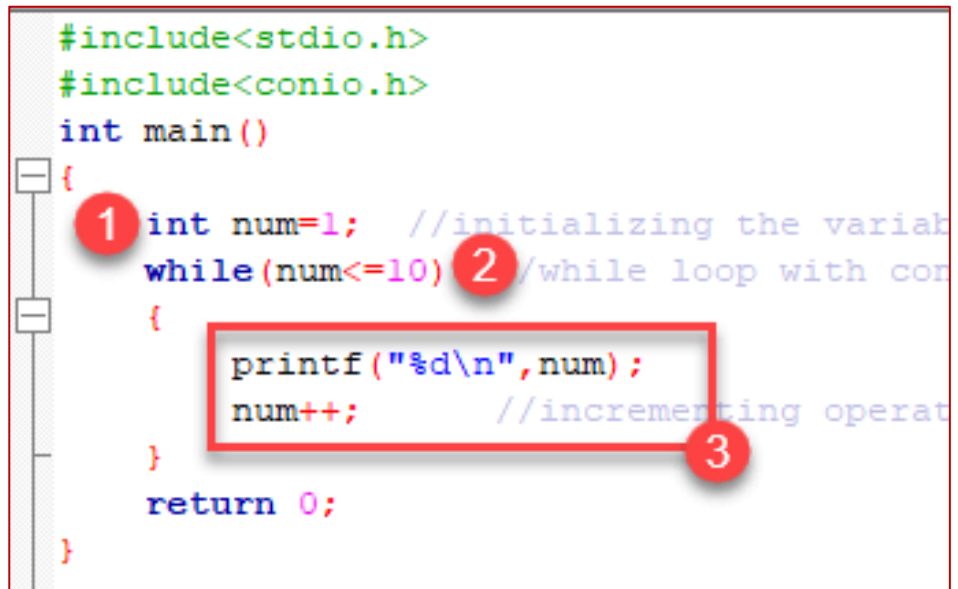
C++ program

```
10 main() {
11
12     Student s1;
13     Student st[100]; //Array of students
14
15     s1.ID = 10;
16     s1.age = 29;
17
18     printf("Student ID: %d\n", s1.ID);
19     printf("Student age: %d\n", s1.age);
20 }
```

LOOP

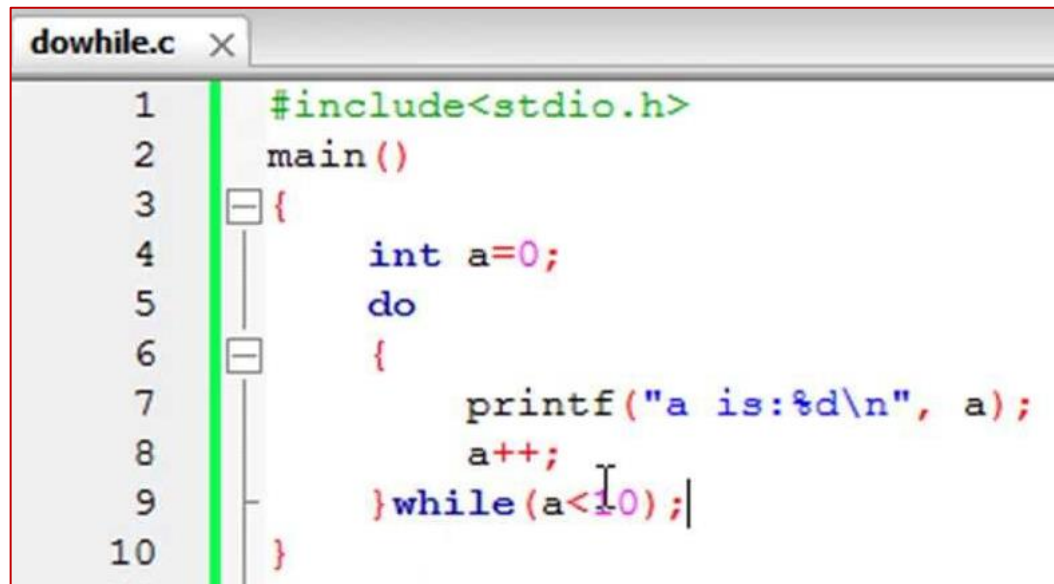
What is the output from the following program?

```
#include<stdio.h>
#include<conio.h>
int main()
{
    1 int num=1; //initializing the variable
    while(num<=10) 2 //while loop with condition
    {
        printf ("%d\n",num);
        num++; //incrementing operation
    }
    return 0;
}
```



LOOP

What is the output from the following program?



```
dowhile.c x
1  #include<stdio.h>
2  main()
3  {
4      int a=0;
5      do
6      {
7          printf("a is:%d\n", a);
8          a++;
9      }while(a<10);
10 }
```

The screenshot shows a code editor window titled 'dowhile.c'. The code is a C program that includes the standard input/output library, defines a main function, and uses a do-while loop to print the value of 'a' (starting at 0) ten times, incrementing 'a' by 1 each time. The loop condition is 'a < 10'.

ARRAY

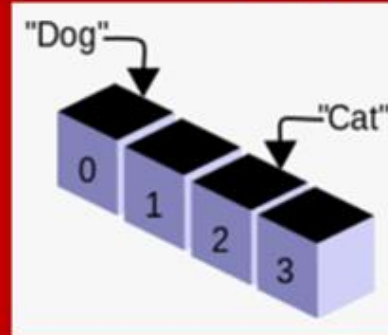
```
int marks[5];
```

80	60	70	85	75
----	----	----	----	----

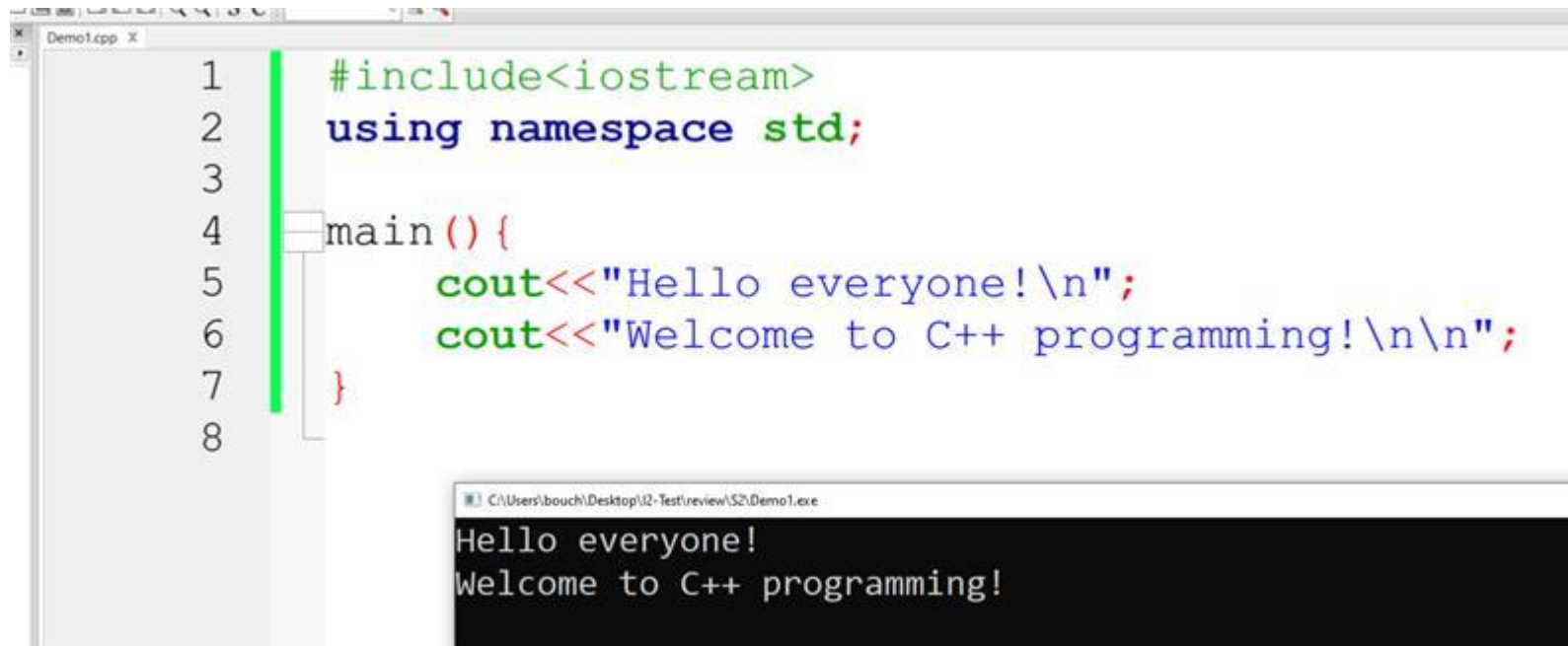
marks[0] marks[1] marks[2] marks[3] marks[4]

Initialization of Array

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



EXAMPLES: C++ PROGRAMMING



The image shows a code editor window titled 'Demo1.cpp' with the following C++ code:

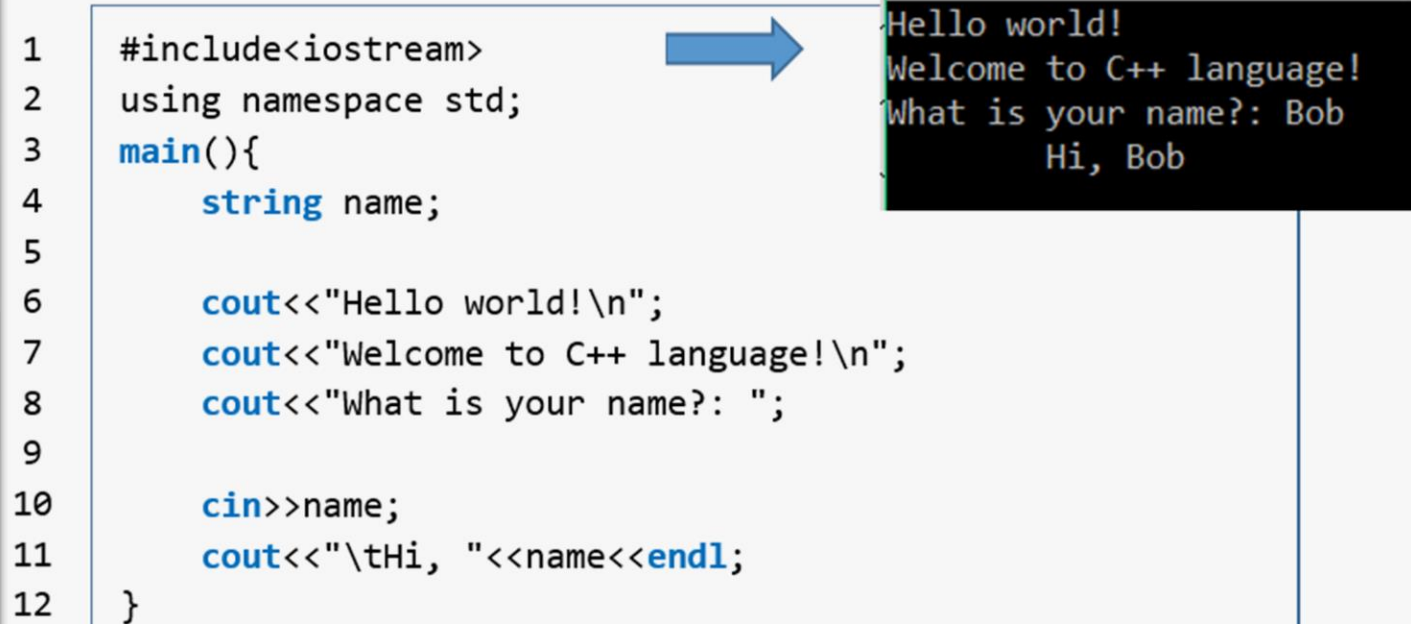
```
1  #include<iostream>
2  using namespace std;
3
4  main() {
5      cout<<"Hello everyone!\n";
6      cout<<"Welcome to C++ programming!\n\n";
7  }
8
```

Below the code editor is a terminal window showing the output of the program:

```
C:\Users\bouch\Desktop\i2-Test\review\S2\Demo1.exe
Hello everyone!
Welcome to C++ programming!
```

1st program in C++ programming

EXAMPLES: C++ PROGRAMMING



The diagram illustrates the execution of a C++ program. On the left, a code editor shows the source code with line numbers 1 through 12. A blue arrow points from the code to a black terminal window on the right, which displays the program's output. The code includes headers, uses the std namespace, and defines a main function that declares a string variable, prints three lines of text, prompts for user input, reads the input, and prints a personalized greeting.

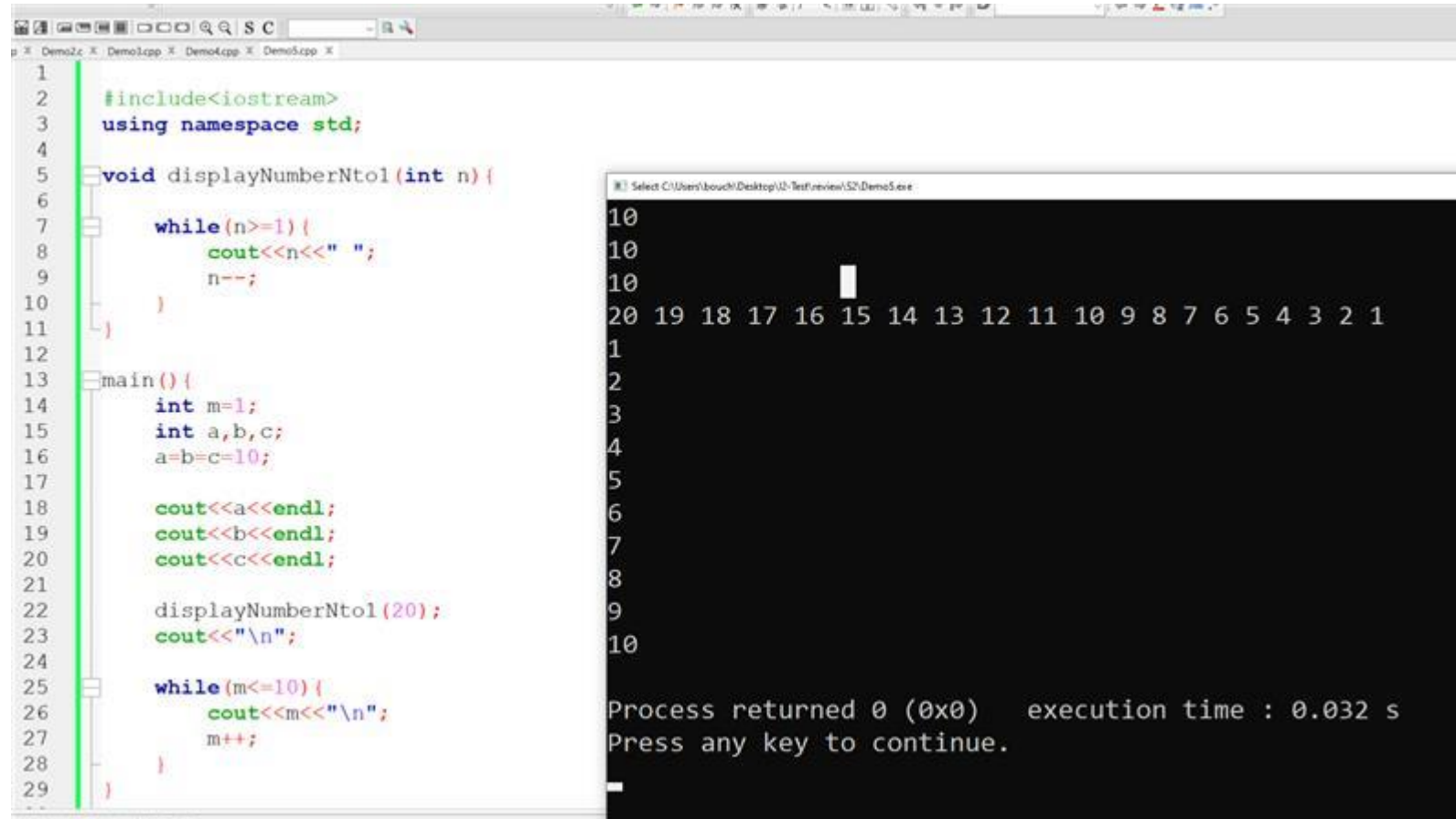
```
1  #include<iostream>
2  using namespace std;
3  main(){
4      string name;
5
6      cout<<"Hello world!\n";
7      cout<<"Welcome to C++ language!\n";
8      cout<<"What is your name?: ";
9
10     cin>>name;
11     cout<<"\tHi, "<<name<<endl;
12 }
```

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



```
1
2  #include<iostream>
3  using namespace std;
4
5  void displayNumberNtol(int n) {
6
7      while(n>=1) {
8          cout<<n<<" ";
9          n--;
10     }
11 }
12
13 main() {
14     int m=1;
15     int a,b,c;
16     a=b=c=10;
17
18     cout<<a<<endl;
19     cout<<b<<endl;
20     cout<<c<<endl;
21
22     displayNumberNtol(20);
23     cout<<"\n";
24
25     while(m<=10) {
26         cout<<m<<"\n";
27         m++;
28     }
29 }
```

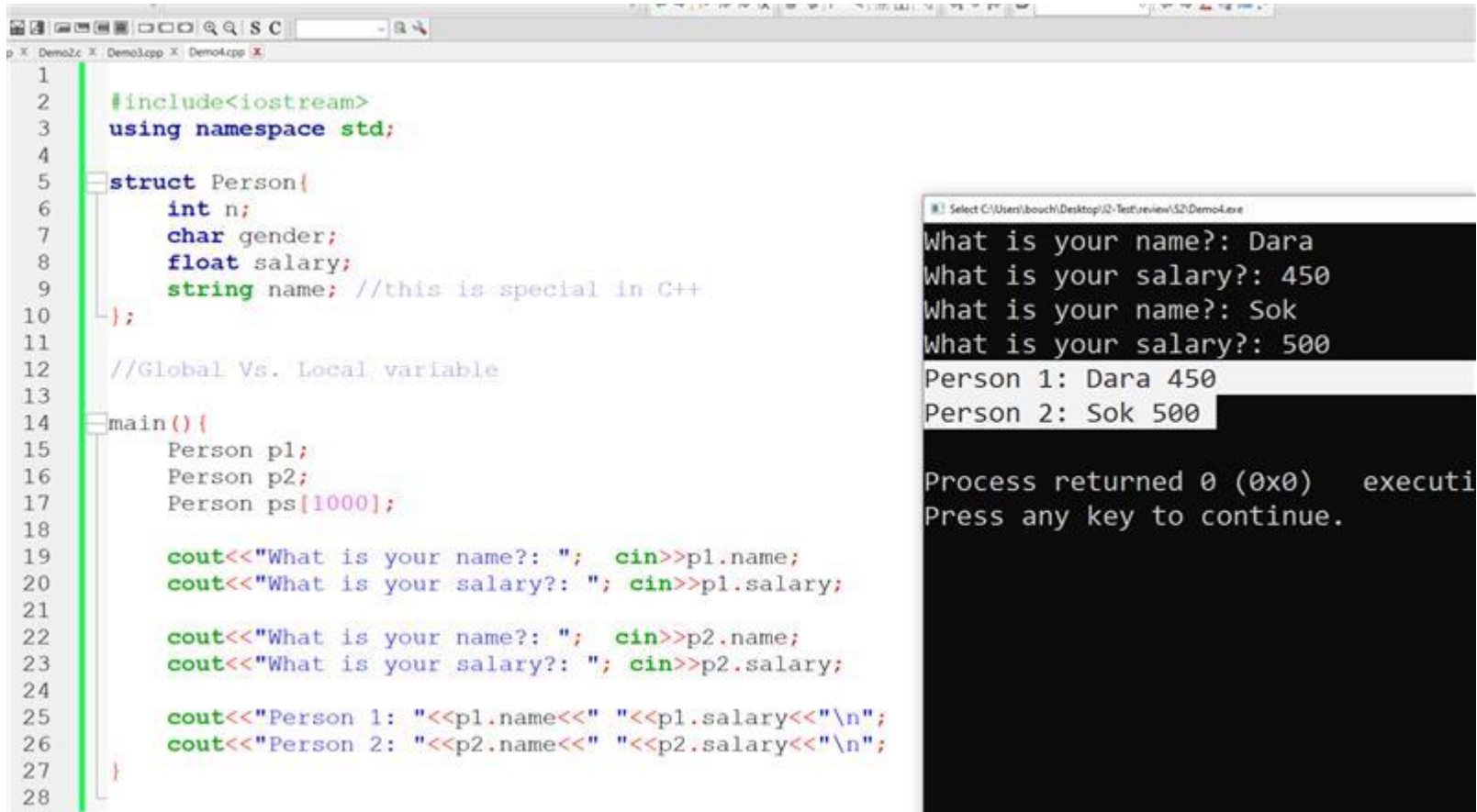
Select C:\Users\bouch\Desktop\12-Test\review\52\Demo5.exe

```
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

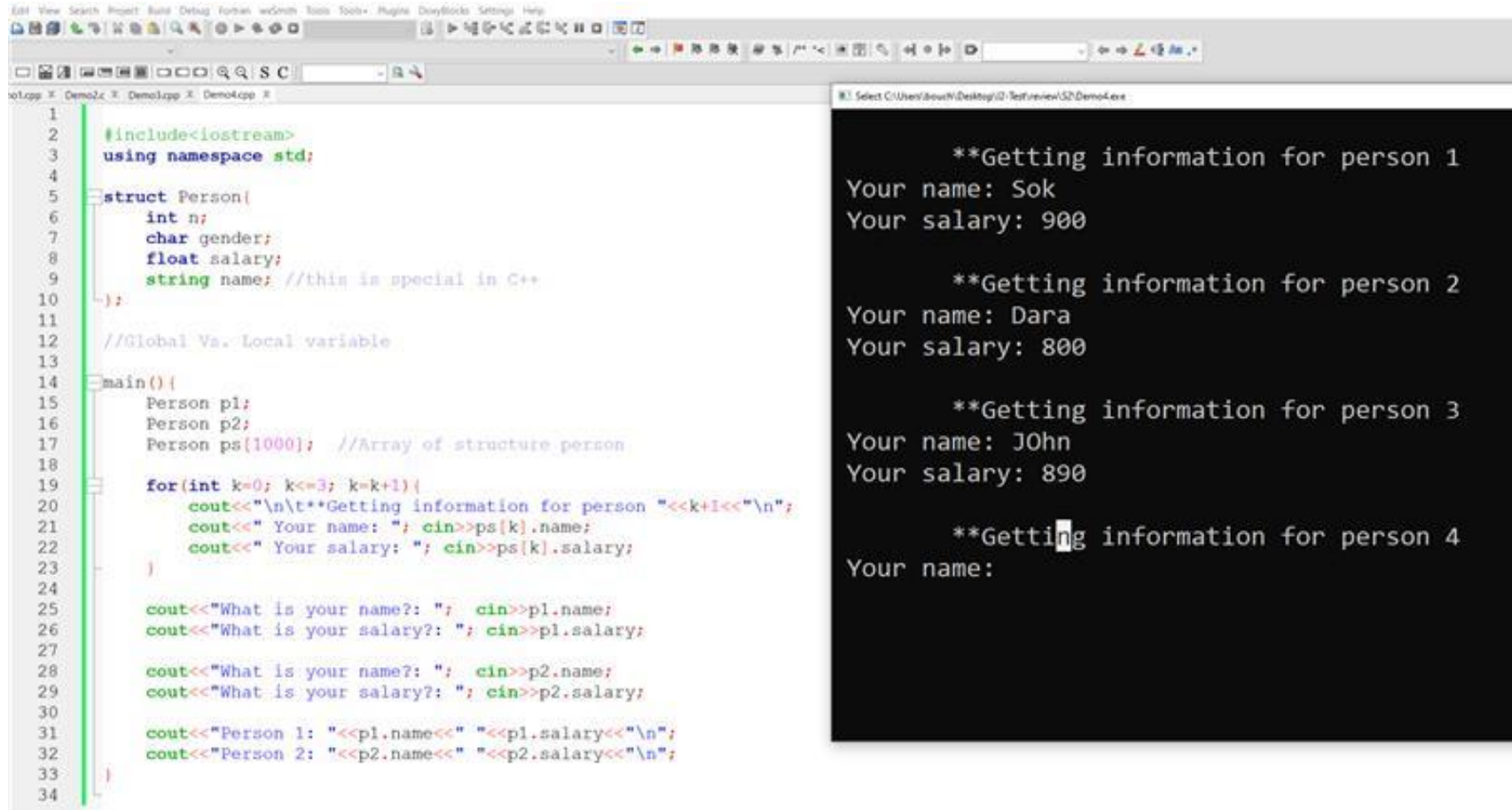


```
1
2 #include<iostream>
3 using namespace std;
4
5 struct Person{
6     int n;
7     char gender;
8     float salary;
9     string name; //this is special in C++
10 };
11
12 //Global Vs. Local variable
13
14 main(){
15     Person p1;
16     Person p2;
17     Person ps[1000];
18
19     cout<<"What is your name?: "; cin>>p1.name;
20     cout<<"What is your salary?: "; cin>>p1.salary;
21
22     cout<<"What is your name?: "; cin>>p2.name;
23     cout<<"What is your salary?: "; cin>>p2.salary;
24
25     cout<<"Person 1: "<<p1.name<<" "<<p1.salary<<"\n";
26     cout<<"Person 2: "<<p2.name<<" "<<p2.salary<<"\n";
27 }
28
```

```
Select C:\Users\bouch\Desktop\2-Test\review\52\Demo4.exe
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



The image shows a screenshot of a C++ IDE with two windows. The left window displays the source code for a program that uses an array of structures. The right window shows the program's output, which prompts for and displays information for four people.

```
1
2 #include<iostream>
3 using namespace std;
4
5 struct Person{
6     int n;
7     char gender;
8     float salary;
9     string name; //this is special in C++
10 };
11
12 //Global Vs. Local variable
13
14 main(){
15     Person p1;
16     Person p2;
17     Person ps[1000]; //Array of structure person
18
19     for(int k=0; k<=3; k=k+1){
20         cout<<"\n\t**Getting information for person "<<k+1<<"\n";
21         cout<<" Your name: "; cin>>ps[k].name;
22         cout<<" Your salary: "; cin>>ps[k].salary;
23     }
24
25     cout<<"What is your name?: "; cin>>p1.name;
26     cout<<"What is your salary?: "; cin>>p1.salary;
27
28     cout<<"What is your name?: "; cin>>p2.name;
29     cout<<"What is your salary?: "; cin>>p2.salary;
30
31     cout<<"Person 1: "<<p1.name<<" "<<p1.salary<<"\n";
32     cout<<"Person 2: "<<p2.name<<" "<<p2.salary<<"\n";
33
34 }
```

Output:

```
**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++