

# 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

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

C++  
syntax

```
main(){
    string name;
    cout<<"Hello world!\n";
    cout<<"What is your name?: ";
    cin>>name;
    cout<<"Hi, "<<name<<endl;
    cout<<"Bye!";
}
```

Filename: test.cpp

Filename: test.c

# MORE ON C AND C++

---

No.	C	C++
1)	C follows the <b>procedural style programming</b> .	C++ is multi-paradigm. It supports both <b>procedural and object oriented</b> .
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 <b>top-down approach</b> .	C++ follows the <b>bottom-up approach</b> .
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, <b>scanf() and printf()</b> are mainly used for input/output.	C++ mainly uses stream <b>cin and cout</b> 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 <b>procedures and modules</b>	C++ programs are divided into <b>functions and classes</b> .

# 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

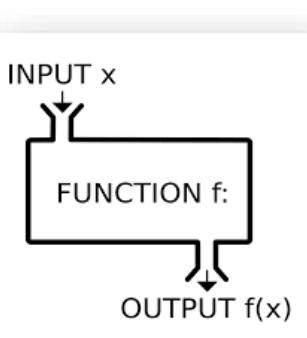
- Function
- Structure
- Loop
- Other

```
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 checkPositive(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



The diagram illustrates the structure of a C-style structure definition. On the left, the code is shown:

```
struct keyword      tag or structure tags
struct bill
{
    float amount;
    int id;
    char address[100];
};
```

Annotations explain the components:

- struct keyword**: Points to the word "struct".
- tag or structure tags**: Points to the identifier "bill".
- Members or Field of Structure**: A brace groups the three members: "float amount", "int id", and "char address[100]".

# 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 operator
    }
    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 }
```

# ARRAY

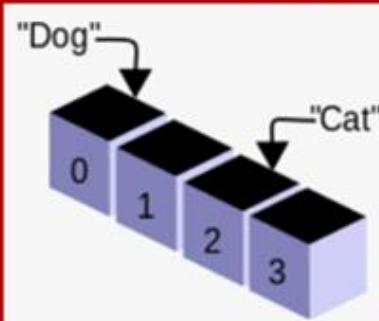
```
int marks[5];
```

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

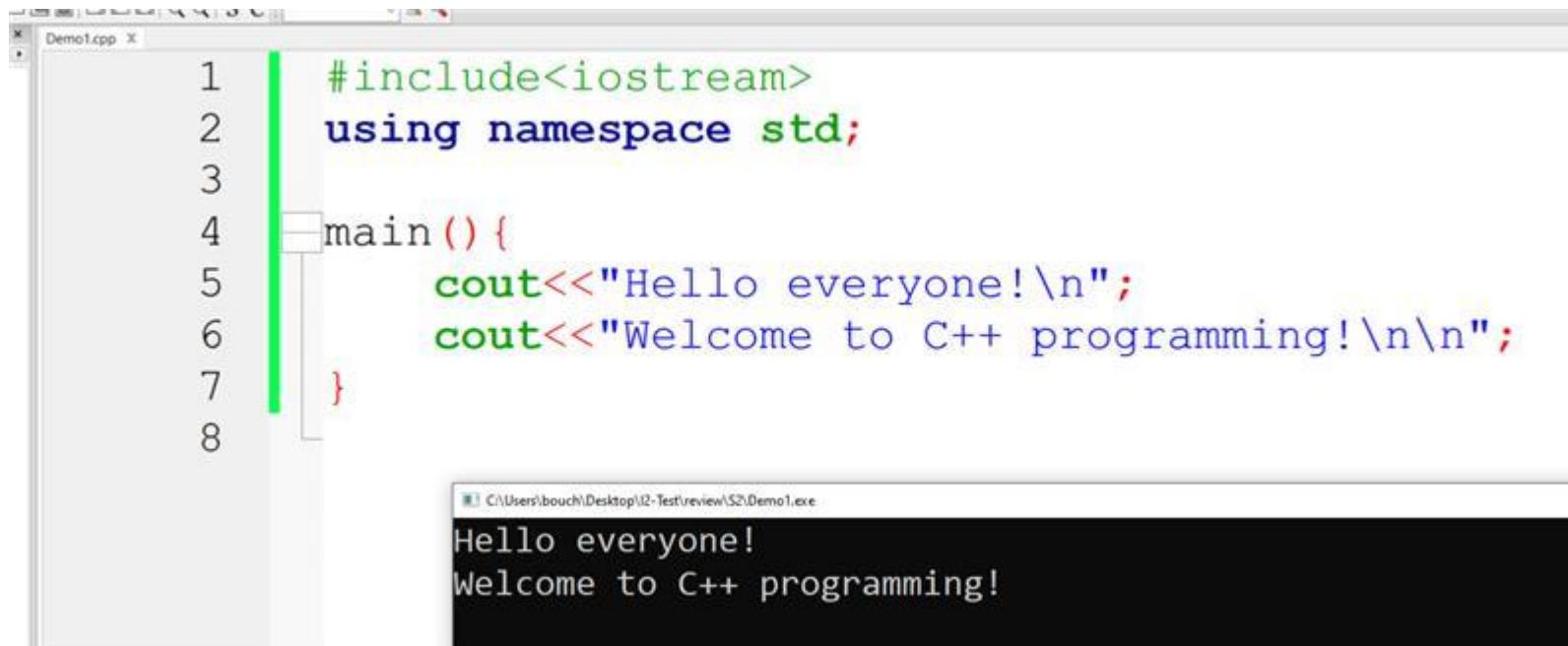
marks[0]    marks[1]    marks[2]    marks[3]    marks[4]

Initialization of Array

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



# EXAMPLES: C++ PROGRAMMING



The screenshot shows a code editor window titled "Demo1.cpp" containing the following C++ code:

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

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

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

```
C:\Users\bouch\Desktop\I2-Test\review\S2\Demo1.exe
Hello everyone!
Welcome to C++ programming!
```

1<sup>st</sup> program in C++ programming

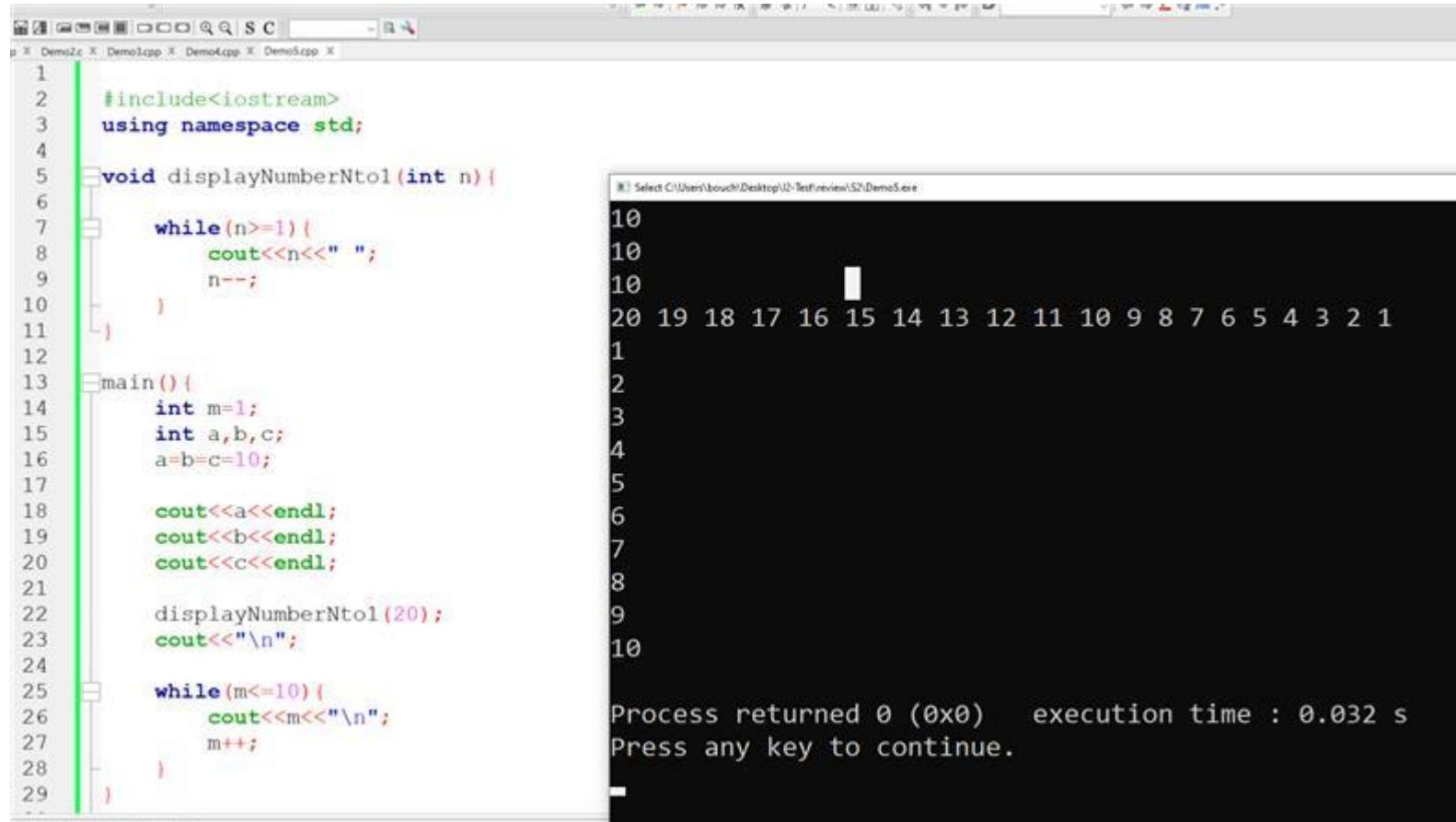
# EXAMPLES: C++ PROGRAMMING

```
1 #include<iostream>           → Hello world!
2 using namespace std;          Welcome to C++ language!
3 main(){                      What is your name?: Bob
4     string name;              Hi, Bob
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 }
```

Figure 1: A basic C++ program overview

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

# EXAMPLES: C++ PROGRAMMING



The image shows a screenshot of a C++ development environment. On the left, the code editor displays a file named Demo5.cpp with the following content:

```
#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++;
    }
}
```

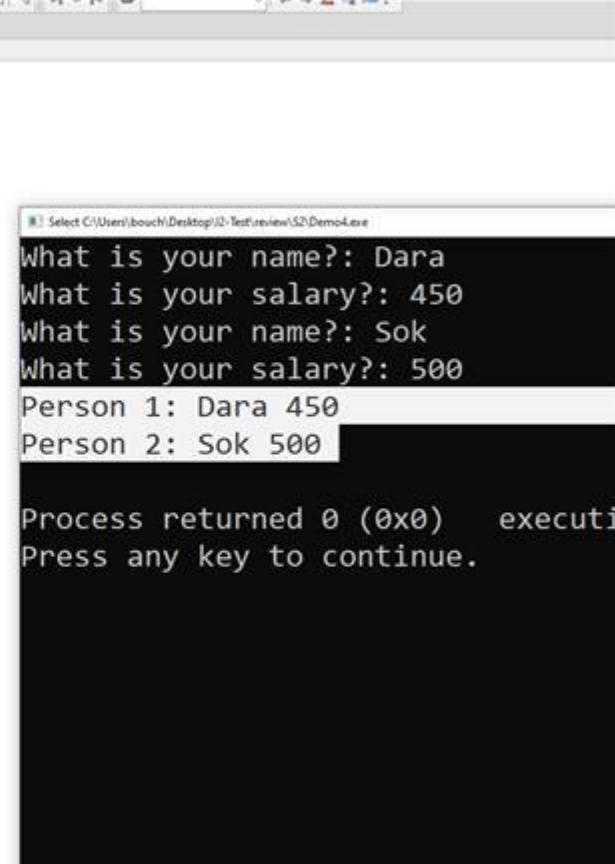
On the right, the terminal window shows the execution of the program. The output is:

```
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++

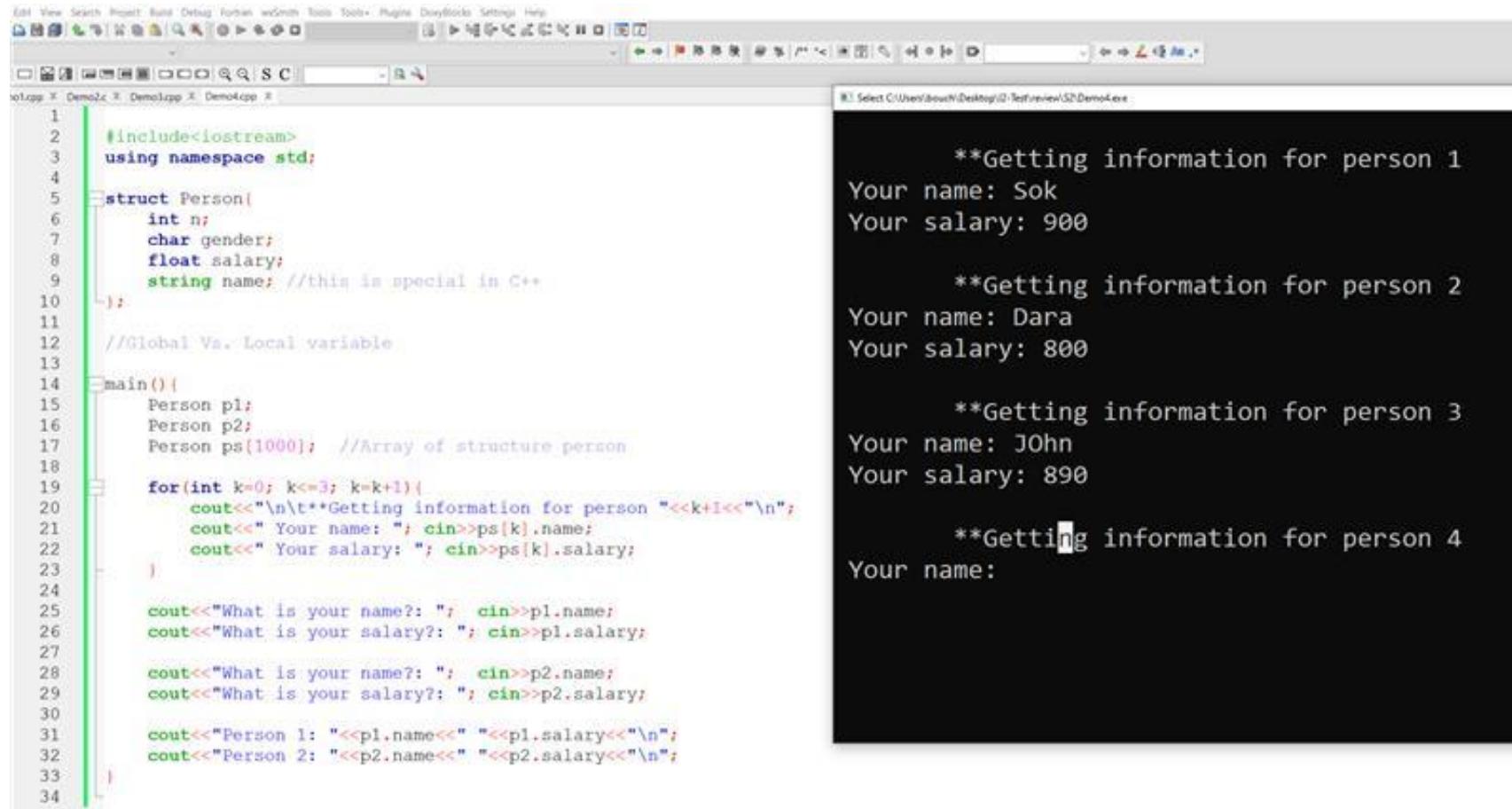
# EXAMPLES: C++ PROGRAMMING



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

Using structure in C++

# EXAMPLES: C++ PROGRAMMING



The image shows a screenshot of a C++ development environment. On the left, the code editor displays a file named 'Demo4.cpp' with the following content:

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

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

//Global Va. 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";
}
```

On the right, a terminal window titled 'Select C:\Users\Aboud\Desktop\Q\Test\review\S2\Demo4exe' shows the output of the program:

```
**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++