

**Mr. Giansante**



**C++ Programming**  
**Fibonacci**

**August 2018**

# Fibonacci Sequence

Source: Wikipedia.org

In mathematics, the Fibonacci numbers are the numbers in the following integer sequence, called the Fibonacci sequence, and characterized by the fact that every number after the first two is the sum of the two preceding ones:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Often, especially in modern usage, the sequence is extended by one more initial term:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

By definition, the first two numbers in the Fibonacci sequence are either 1 and 1, or 0 and 1, depending on the chosen starting point of the sequence, and each subsequent number is the sum of the previous two.

The sequence  $F_n$  of Fibonacci numbers is defined by the recurrence relation:

$$F_n = F_{n-1} + F_{n-2}$$

with  $F_1 = 1$  and  $F_2 = 1$  -or-  $F_1 = 0$  and  $F_2 = 1$ .

The Fibonacci spiral: an approximation of the golden spiral created by drawing circular arcs connecting the opposite corners of squares in the Fibonacci tiling; the one to the right uses squares of sizes 1, 1, 2, 3, 5, 8, 13 and 21.

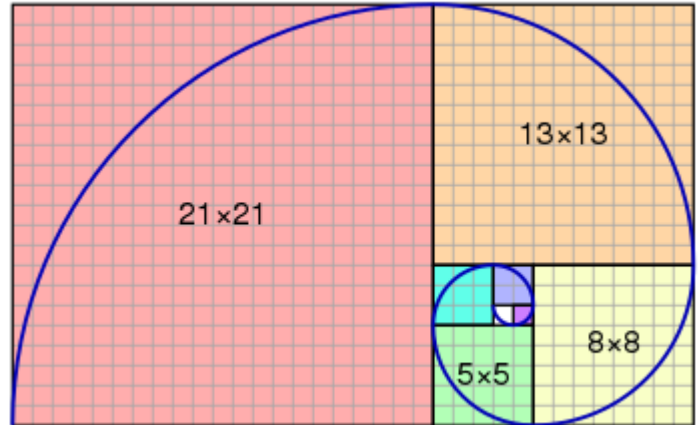


Image: by Jahobr - Own work, CC0,  
<https://commons.wikimedia.org/w/index.php?curid=58460223>

# Fibonacci Sequence with three Variables

---

```
#include <iostream>
using namespace std;
int main()
{
    int t1 = 1;
    int t2 = 1;
    int t3;

    cout << "Fibonacci sequence: 1, 1";

    for(int i = 1; i < 21; i++)
    {
        t3 = t1 + t2;
        cout << ", " << t3;

        t1 = t2;
        t2 = t3;
    }

    return 0;
}
```

# Fibonacci Sequence with an Array

---

```
#include <iostream>
using namespace std;
int main()
{
    int fibonacci[20];

    fibonacci[1] = 1;
    fibonacci[2] = 1;

    // Place terms into Array
    for(int n = 3; n < 20; n++)
    {
        fibonacci[n] = fibonacci[n-2] + fibonacci[n-1];
    }

    // Print Array
    cout << "Fibonacci sequence: ";
    for(int n = 1; n < 20; n++)
    {
        cout << fibonacci[n] << ", ";
    }

    return 0;
}
```